09/341633 6

PCT/JP98/00233

22.01.98

日 本 国 特 許 庁
PATENT OFFICE
JAPANESE GOVERNMENT

別紙添付の書類に記載されている事項は下記の出願書類に記載されている事項と同一であることを証明する。

This is to certify that the annexed is a true copy of the following application as filed with this Office.

出願年月日
Date of Application: １９９７年 １月２３日

出願番号
Application Number: 平成 ９年特許願第０１０５９２号

出願人
Applicant (s): シャープ株式会社

１９９８年 ３月 ６日

特許庁長官
Commissioner,
Patent Office

荒井寿光

出証番号 出証特平１０－３０１２６４７

【書類名】　特許願

【整理番号】　96-01454

【提出日】　平成 9年 1月23日

【あて先】　特許庁長官　殿

【国際特許分類】　G06F　3/14

G06F　3/153

【発明の名称】　プログラマブル表示装置

【請求項の数】　9

【発明者】

【住所又は居所】　大阪府大阪市阿倍野区長池町２２番２２号　シャープ株式会社内

【氏名】　中村　聡

【発明者】

【住所又は居所】　大阪府大阪市阿倍野区長池町２２番２２号　シャープ株式会社内

【氏名】　山村　博幸

【発明者】

【住所又は居所】　大阪府大阪市阿倍野区長池町２２番２２号　シャープ株式会社内

【氏名】　山本　真司

【発明者】

【住所又は居所】　大阪府大阪市阿倍野区長池町２２番２２号　シャープ株式会社内

【氏名】　守屋　政明

【特許出願人】

【識別番号】　000005049

【郵便番号】　545

【住所又は居所】　大阪府大阪市阿倍野区長池町２２番２２号

【氏名又は名称】　シャープ株式会社

【代表者】　辻　晴雄

【代理人】

【識別番号】　100069534

【弁理士】

【氏名又は名称】　藤本　博光

【電話番号】　03(3593)2361

【手数料の表示】

【予納台帳番号】　007951

【納付金額】　21,000円

【提出物件の目録】

【物件名】　明細書　1

【物件名】　図面　1

【物件名】　要約書　1

【包括委任状番号】　9106003

【書類名】　　明細書

【発明の名称】　プログラマブル表示装置

【特許請求の範囲】

【請求項１】　表示データが格納されているメインメモリと、

前記表示データのデータ形式を画面表示のデータ形式に変換するデータ処理回路部と、

前記データ処理回路部に変換された表示データを表示ライン単位に格納する複数のラインメモリと、

前記メインメモリから前記ラインメモリに表示データを転送格納させて、該ラインメモリから必要な表示データを読み出して画面表示させる制御を行う表示制御部と、

前記メインメモリに前記表示データを格納させ、データ形式及び格納アドレスを含む格納情報を前記表示制御部に転送する主制御部と、

を備え、

前記表示制御部は、転送元の前記メインメモリに対し前記格納情報に基づいて、画面表示させる可能性のある１ライン分の表示データのアドレスを指定して該表示データを読み出し、前記データ処理回路部にデータ変換をさせて、前記ラインメモリを選択し該表示データを格納させることを特徴とするプログラマブル表示装置。

【請求項２】　前記表示制御部は、前記ラインメモリに繰り返し利用する表示データを格納し、該繰り返し表示データを表示させる場合には、前記ラインメモリから前記繰り返し表示データのアドレスを指定して読み出し、画面表示させることを特徴とする請求項１記載のプログラマブル表示装置。

【請求項３】　繰り返し利用する表示データを格納するデータバッファメモリを備え、

前記表示制御部は、前記データを画面表示させる場合、前記データバッファメモリから前記繰り返し表示データを読み出し、画面表示させることを特徴とする請求項１記載のプログラマブル表示装置。

【請求項４】　前記メインメモリから読み出した表示データを格納する第１バッファメモリと、

前記第１バッファメモリから読み出した表示データを格納する第２バッファメモリと、

前記第１及び第２バッファメモリの読み出し及び書き込みアドレスをカウントするアドレスカウンタと、

を備え、

前記表示制御部は、前記アドレスカウンタに対し読み出し及び書き込みアドレスカウントをそれぞれ停止／動作の制御を行い、拡大・縮小・スキップの処理を行って、そのデータを前記ラインメモリに格納することを特徴とする請求項１記載のプログラマブル表示装置。

【請求項５】　前記表示制御部は、第１バッファメモリからの読みだしアドレスカウントを所定の順に停止／動作を繰り返させることを特徴とする請求項４記載のプログラマブル表示装置。

【請求項６】　前記データ処理回路部は、各種のデータ形式を変換する複数の変換処理回路を有し、

前記表示制御部は、前記格納情報のデータ形式情報に基づいて前記変換処理回路を選択することを特徴とする請求項１記載のプログラマブル表示装置。

【請求項７】　前記表示制御部に必要なプログラムとデータを格納するプログラムメモリとデータメモリとを備えたことを特徴とする請求項１記載のプログラマブル表示装置。

【請求項８】　前記表示制御部は、前記プログラムメモリと前記データメモリに必要な情報を前記メインメモリから転送させることを特徴とする請求項７記載のプログラマブル表示装置。

【請求項９】　前記表示制御部は、前記ラインメモリに表示データを格納する際に何ライン目で使用するデータであるかを示すライン情報を付加し、前記ラインメモリから表示データを読み出す際にライン情報も同時に読み出して、該表示データを使用するラインがライン情報と同一である場合のみ画面表示させることを特徴とする請求項１記載のプログラマブル表示装置。

【発明の詳細な説明】

【０００１】

【発明の属する技術分野】

この発明は映像データを表示するコンピュータ装置におけるプログラマブル表示装置に関し、特にグラフィックディスプレイシステムにおけるメモリからの表示用データの読み出し方が極めて柔軟であり、メモリから表示用データを読み出す際に読み出す画素データの最小単位を画素毎として、動的に定義できるシステムに関する。

【０００２】

【従来の技術】

従来、一般的なコンピュータにおいて、表示データの重ね合わせや合成処理は、単一のフレームメモリ上において、そのメモリ上で直接メインプロセッサあるいは描画装置が演算して合成する。図２７は、従来の画像表示装置の一例を示すブロック図である。この画像表示装置は、メインＣＰＵ１０１、メインメモリ１０２、データ処理回路１０３、ラインメモリ１０４、出力処理回路１０５、システムコントローラ１０６、同期信号生成回路１０７からなる構成である。

【０００３】

メインメモリ１０２には、表示データがいくつか格納されている。例えば、数種類のウィンドウ表示を行う場合を考えると、各ウィンドウに対応した表示データが格納されている。このウィンドウを重ね合わせて一画面に表示させる場合、一画面表示になるように、メインＣＰＵ１０１が各表示データを選択して読み出し、一画面の表示データを再びメインメモリ１０２に格納する。同期信号生成回路１０７で発生した同期信号のタイミングに従って、システムコントローラ１０６がデータ転送用にメインメモリ１０２のアドレスを生成する。このアドレスに従ってメインメモリ１０２から表示データを読み出して、あらかじめ決められたデータ処理をデータ処理回路１０３で行った後、ラインメモリ１０４にデータを転送する。ラインメモリ１０４からのデータは同期信号のタイミングにしたがって出力され、出力処理回路１０５により表示用の処理を行ってディスプレイに表示する。

【０００４】

また、特開平６－１４９５２７号公報に開示されているように、重ねあわせに必要な枚数分フレームメモリを用意して、映像出力時に全部のフレームメモリからデータを読み出し各フレーム間の優先順位を元に合成結果を表示するシステムがある。

【０００５】

また、特開平６－２９５１６９号公報に開示されているように、表示用のメモリとは別に表示領域のメモリの各表示ドット毎に識別メモリを設けることで各表示ドットが今どのようなモード（例えば１画素のビット数）にあるかを識別し、そのモードに合わせて表示し、１つの画面上で異なる表示モードを表示するシステムがある。

【０００６】

また、前記特許にもあるように識別メモリの内容を参照したり、特開平７－３４３４２号公報に開示するように別途マスクメモリを利用し、表示している各ウインドウ内の情報を変更して書き換える場合にその領域外をマスクするシステムがある。

【０００７】

【発明が解決しようとする課題】

しかしながら従来のようにメインＣＰＵ１０１が、各ウインドウの重ね合わせ等の処理を行う場合には、メインＣＰＵ１０１の負担が過大となり、他の処理を行うことができずに全体の処理速度が遅くなるなどの問題があった。

【０００８】

また、各ウインドウの重ねあわせをするのに必要な枚数分のフレームメモリを持つことで、ソフトウエアの処理負荷を軽減する方法では、始めからそのシステムで必要と思われる最大枚数分のフレームメモリが必要である。つまり、画面上に表示するウインドウのサイズに関係なく表示エリアの最大サイズでフレームメモリを必要とする。そのため、メモリの利用効率が極めて悪くなる上に、多数のウインドウを同時に開いた場合、そのウインドウに対応する全てのフレームメモリから同時にデータを読み出す必要がある。すなわち、ウインドウが重なってお

り実際は表示されないような部分のデータも読み出す必要がある。こうして、画面上に開くウインドウの枚数に比例して消費電力が大きくなってしまう。

【0009】

また、従来の様に１つの画面上で異なる表示モードを混在表示する方法として表示領域のメモリの各表示ドット毎に識別メモリを設けることで各表示ドットが今どのようなモードにあるかを識別する方法がある。その方法では、フルスクリーン分のメモリに対して別途数ビットの識別メモリが必要となるため、別の用途に転用することのできないメモリ（識別メモリ）が余分に必要となる。このことは、マスクメモリを利用する場合にも同様なことが言える。

【0010】

本発明の目的は、表示データを収納するメモリ空間を必要なものだけとし、表示のためのメモリアクセス回数を押さえて処理を高速化できるとともに、主制御部の負担を軽減することができるプログラマブル表示装置を提供することである。

【0011】

【課題を解決するための手段】

請求項１の発明は、表示データが格納されているメインメモリと、前記表示データのデータ形式を画面表示のデータ形式に変換するデータ処理回路部と、前記データ処理回路部に変換された表示データを表示ライン単位に格納する複数のラインメモリと、前記メインメモリから前記ラインメモリに表示データを転送格納させて、該ラインメモリから必要な表示データを読み出して画面表示させる制御を行う表示制御部と、前記メインメモリに前記表示データを格納させ、データ形式及び格納アドレスを含む格納情報を前記表示制御部に転送する主制御部と、を備えたプログラマブル表示装置である。前記表示制御部は、転送元の前記メインメモリに対し前記格納情報に基づいて、画面表示させる可能性のある１ライン分の表示データのアドレスを指定して該表示データを読み出し、前記データ処理回路部にデータ変換をさせて、前記ラインメモリを選択し該表示データを格納させることを特徴とする。

【００１２】

請求項２の発明は、前記表示制御部は、前記ラインメモリに繰り返し利用する表示データを格納し、該繰り返し表示データを表示させる場合には、前記ラインメモリから前記繰り返し表示データのアドレスを指定して読み出し、画面表示させることを特徴とする請求項１記載のプログラマブル表示装置である。

【００１３】

請求項３の発明は、繰り返し利用する表示データを格納するデータバッファメモリを備え、前記表示制御部は、前記データを画面表示させる場合、前記データバッファメモリから前記繰り返し表示データを読み出し、画面表示させることを特徴とする請求項１記載のプログラマブル表示装置である。

【００１４】

請求項４の発明は、前記メインメモリから読み出した表示データを格納する第１バッファメモリと、前記第１バッファメモリから読み出した表示データを格納する第２バッファメモリと、前記第１及び第２バッファメモリの読み出し及び書き込みアドレスをカウントするアドレスカウンタとを備える請求項１記載のプログラマブル表示装置である。前記表示制御部は、前記アドレスカウンタに対し読み出し及び書き込みアドレスカウントをそれぞれ停止／動作の制御を行い、拡大・縮小・スキップの処理を行って、そのデータを前記ラインメモリに格納することを特徴とする。

【００１５】

請求項５の発明は、前記表示制御部が、データ第１バッファメモリからの読みだしアドレスカウントを所定の順に停止／動作を繰り返すことを特徴とする請求項４記載のプログラマブル表示装置である。

【００１６】

請求項６の発明は、前記データ処理回路部は、各種のデータ形式を変換する複数の変換処理回路を有し、前記表示制御部は、前記格納情報のデータ形式情報に基づいて前記変換処理回路を選択することを特徴とする請求項１記載のプログラマブル表示装置である。

【００１７】

請求項７の発明は、前記表示制御部に必要なプログラムとデータを格納するプログラムメモリとデータメモリとを備えたことを特徴とする請求項１記載のプログラマブル表示装置である。

【００１８】

請求項８の発明は、前記表示制御部は、前記プログラムメモリと前記データメモリに必要な情報を前記メインメモリから転送させることを特徴とする請求項７記載のプログラマブル表示装置である。

【００１９】

請求項９の発明は、前記表示制御部は、前記ラインメモリに表示データを格納する際に何ライン目で使用するデータであるかを示すライン情報を付加し、前記ラインメモリから表示データを読み出す際にライン情報も同時に読み出して、該表示データを使用するラインがライン情報と同一である場合のみ画面表示させることを特徴とする請求項１記載のプログラマブル表示装置である。

【００２０】

請求項１の発明において、表示をする際に必要な部分の表示データをメインメモリ内から取り出して使用する。そのため、メインメモリ内の任意の位置のデータを取り出して任意に組み合わすことが可能である。この制御はすべて表示制御部が行い、主制御部が処理を行う必要がなく、主制御部のソフトウエアにおける処理負荷を低減できる。

【００２１】

請求項２の発明において、ウインドウシステムにおける背景等のように、ライン方向に対して繰り返すようなデータであった場合、読み出しラインメモリアドレスを任意の位置でループできる。

【００２２】

請求項３の発明において、カーソルや繰り返し背景などをデータバッファメモリに収納しておけるため、決まりきったデータをメインメモリから読み出す必要が無い。データバスの使用回数を減らすことができる。

【0023】

請求項4の発明において、表示データを読み出す際に拡大縮小処理をするため、表示用データに対する拡大縮小処理を事前にする必要が無く、バスの使用効率を上げられる。また、ビデオ入力映像を表示する場合に映像サイズの変更が必要となるのが常であるが、出力段に拡大縮小処理を掛けることで拡大縮小回路がより有効に利用できる。また、この事によりビデオデータを常にフルサイズで取り込みながら、そのデータを一旦フレームメモリなどに転送することなく表示は任意のサイズに設定できる。

【0024】

請求項5の発明において、第1バッファメモリからの読みだしアドレスカウントを所定の順に停止／動作を繰り返すことにより、一定倍率の拡大・縮小が簡単な処理で行うことができる。

【0025】

請求項6の発明において、表示制御部は、格納情報のデータ形式情報に基づいてデータ変換ができるので、表示用データを収納する形式などに制限が無い。

【0026】

請求項7の発明において、前記表示制御部に必要なプログラムとデータを格納するプログラムメモリとデータメモリとを備えるので、処理の度にメインメモリからデータを読み出す必要がない。

【0027】

請求項8の発明において、前記表示制御部は、前記プログラムメモリとデータメモリに必要な情報をメインメモリから転送させるため、画面モードあるいはグラフィック領域の変更に柔軟に対応できる。容量を越えたプログラムあるいはデータは、メインメモリから読み出せばよいので、メモリの容量は小さくて済む。

【0028】

請求項9の発明において、各ラインの表示毎にラインメモリ内のデータを消去する必要がなく、垂直帰線期間毎にすべてのラインメモリの使用ライン情報を消去するだけでよいので処理の高速化を図れる。

【００２９】

【発明の実施の形態】

以下に、本発明の実施形態について図面を用いて説明する。

図１は、本発明に係るプログラマブル表示装置の一実施形態を示すブロック図である。この表示装置は、メインＣＰＵ１１、プログラムや表示データやその他のデータを記憶するメインメモリ１２、メインメモリ１２の表示データをディスプレイ表示のデータ形式に変換する処理を行うデータ処理回路１３、変換処理された表示データを記憶する表示メモリ部１４、表示データを画面に出力するための処理を行う出力処理回路１７、メインメモリ１２へのデータアクセスを行うＤＭＡ（Direct Memory Access）１８、プログラムメモリ１９、データメモリ２０、プログラムメモリ１９やデータメモリ２０に記述された命令・データを解釈し、それに従っておもに表示データの転送等を行う表示プロセッサー２１、同期信号生成回路２２、ビデオ入力２３，２４とから構成される。

【００３０】

データ処理回路１３は、図２に示すように表示プロセッサ１９より送られてきた表示データに対してＹＵＶ→ＲＧＢ変換を行うＹＵＶデコーダ２７ａ、同表示データに対してランレングス展開を行うランレングス展開回路２７ｂ、同表示データに対してカラーデータの伸長を行うカラー伸長回路２７ｃ、同表示データに対してパレット変換を行う複数のカラーパレット２７ｄ，２７ｅの複数の処理回路と、セレクタ２８とからなる。表示メモリ部１４は、図２に示すように、カーソルのパターンデータなどの格納に使用できるデータバッファ１５と、データ表示データおよび使用ライン情報を記憶する複数のラインメモリ１６とからなる。出力処理回路１７は、複数のラインメモリ１６より任意のラインメモリを選択するセレクタ、αブレンディングを実現するため表示データの明るさを変化させるアッテネータおよびその出力を加算する加算器、繰り返し背景データやカーソルなどの合成に使用されるセレクタ、ディスプレイに表示するためＤ／Ａ変換を行うＤ／Ａコンバータ等からなる。表示プロセッサ２１は、図３に示すように、転送用バッファメモリ２５ａ，２５ｂ，２６ａ，２６ｂを有する。

【００３１】

この表示装置は、専用のフレームバッファを持たず、メインメモリ１２に表示データを同居させるＵＭＡ（Unified Memory Architecture)構成を取り入れるが、メインメモリ１２に専用のフレームバッファを持つ構成になってもかまわない。

【００３２】

以下、この実施形態の動作を説明する。

まず、表示データが実際に表示されるまでのおおまかな流れを以下に説明する。メインＣＰＵ１１により表示データは主にメインメモリ１２に格納されている。これらの表示データは、ＤＭＡ１８によって読み出され、図３に示す表示プロセッサ２１内部の転送用バッファメモリ２５ａ，２５ｂに一時的に格納される。そこで拡大・縮小・スキップなどの操作をされて転送用バッファメモリ２６ａ，２６ｂに格納されたのち、データ処理回路１３により単純なＲＧＢ形式のデータに変換され、ラインメモリ１６に格納される。ラインメモリ１６に書き込まれたデータは、同期信号生成回路２２が発生する同期信号のドットクロックに合わせて１画素分ずつ読み出される。出力処理回路１７によって２画面のαブレンディング処理、又は繰り返し背景データやカーソルなどと合成され、Ｄ／Ａ変換されて同期信号などと共にディスプレイに出力されて表示される。以上が表示までのおおまかな流れである。

【００３３】

この表示装置において、表示のかなりの部分の制御は表示プロセッサ２１よって行われる。表示プロセッサ２１は専用のプログラムメモリ１９およびデータメモリ２０を持っており、それに格納されているプログラムおよびデータを解釈して、それに従って表示データの転送等を行う。プログラムメモリ１９およびデータメモリ２０の情報は、必要に応じてメインメモリ１２から転送されてくる。メインメモリ１２には表示構成、グラフィック領域の変更などに応じて複数のプログラム／データを格納しておく。

【００３４】

メインメモリ１２からの表示データの転送命令は、メインＣＰＵ１１から直接

表示プロセッサ２１に対し発行する場合と、表示プロセッサ２１自身が発行する場合がある。転送命令をメインＣＰＵ１１が発行するのは、主に表示モード（１画素の情報を示すビット数）が変更になった場合であり、表示プロセッサ２１自身が発行するのは、主に１画面を構成するために必要なプログラム／データが表示プロセッサのＲＡＭ容量より大きい場合である。このとき表示の途中でプログラム／データを入れ換える。

【００３５】

このような構成にすることで、必要に応じて情報を転送すればよいため、画面モードあるいはグラフィック領域の変更に柔軟に対応できる。かつメモリ容量を越えたプログラムあるいはデータを実行可能とする。こうして表示プロセッサのＲＡＭは小容量で済む。コンパクトあるいは低コストでシステムを構築できる。

【００３６】

また表示プロセッサ２１の動作が決まっていて変更が必要ない場合は、プログラムメモリ１９またはデータメモリ２０はＲＯＭでもかまわない。この場合は、メインメモリ１２から転送するには及ばない。ＲＯＭは同容量のＲＡＭに比べてチップ面積が小さくて済むため、コスト面で有利になる。

【００３７】

次に、表示プロセッサ２１にプログラムを与えて画面表示を行うときの、表示プロセッサ２１の基本動作について説明する。図４（Ａ）～（Ｃ）は、メインメモリ１２の表示データとディスプレイの表示出力を示す説明図である。いずれもあらかじめメインメモリ１２上に格納してある表示データをラインメモリ１６に格納するものである。ベタスクリーンを表示する場合と複数のウインドウ等を合成して表示する場合について説明する。

【００３８】

ベタスクリーンとは、図４（Ａ）に示すようにメインＣＰＵ１１により、背景、カーソル、ウインドウ等を、合成されたベタスクリーンデータとして、あらかじめメインメモリ１２上に格納しておく。表示するには格納されている先頭アドレスから順に読み出してラインメモリ１６に転送していけばよい。

【００３９】

複数ウインドウ等の合成には、αブレンディングを考慮しない場合（図４（Ｂ）参照）と、考慮する場合（図４（Ｃ）参照）の２つの場合がある。αブレンディングとは半透明合成のことで、例えば２つのウインドウが重なる場合、通常ならば重なった部分は手前のウインドウだけが表示されるが、αブレンディングの指定をすると手前のウインドウが透けて、奥のウインドウが見えるようになる。言い換えると、αブレンディングとは、このように複数の表示データをある割合で合成して表示する機能のことを言う。一方、表示プロセッサ２１の動作には実際には拡大・縮小・スキップの操作やデータ処理回路１３や使用ライン情報などの制御が関わってくるが、それらの動作については後述する。

【００４０】

次に、これら画面表示を行う場合の表示装置の動作について説明する。図５は、ベタスクリーンデータを１画面分表示するフローチャートである。図６はそのときの表示画面例であり、図７はベタスクリーンデータが格納されているメインメモリ１２のメモリマップである。まず、１画面を表示するために、ステップＡ１において、ベタスクリーンデータのＸ方向の座標サイズｘｌ、ベタスクリーンデータのＹ方向の座標サイズｙｌを取得する。

【００４１】

次に、ステップＡ２にて、メインメモリ１２上のベタスクリーンデータ格納先頭アドレスｂｅｔａ＿ａｄｄｒをライン番号Ｌに相当するベタスクリーン格納先頭アドレスａｄｄｒとして取得する。これらのデータは、固定データであればプログラム内で固定されたイミディエイトデータとして取得することもできる。また、任意のサイズであればメインメモリ１２上に存在するこれらのデータをデータメモリ２０に転送し、データメモリ２０を参照することによっても取得することが可能である。ステップＡ３にて、現在表示中の水平ライン番号の次の水平ライン番号をライン番号Ｌとして取得し、この値が偶数か奇数かをステップＡ４にて判別する。偶数であれば、ステップＡ５にてラインメモリ１６ａへ、奇数であれば、ステップＡ６にてラインメモリ１６ｂへ、メインメモリ１２のライン番号Ｌに相当するベタスクリーンデータ格納先頭アドレスａｄｄｒからｘｌサイズの

データ転送を行う。

【００４２】

偶数と奇数のライン番号によりラインメモリ１６ａとラインメモリ１６ｂへの書き込みを切り替えているのは、ラインメモリが表示側でアクセスされているとき、表示プロセッサ２１からはアクセスできないためである。表示に使用されているラインメモリとは別のラインメモリを設けることにより、表示中であっても表示プロセッサ１２がラインメモリへアクセスできるようにしている。

【００４３】

ラインメモリ１６ａまたはラインメモリ１６ｂへのデータ転送後、ステップＡ７にて、ベタスクリーンＹ方向の座標サイズｙ１と、次に表示するライン番号Ｌとを比較する。（Ｌ＋１）の値がｙ１よりも小さいときは、ステップＡ８にてライン番号Ｌに相当するベタスクリーンデータ格納先頭アドレスａｄｄｒをベタスクリーンＸ方向の座標サイズｘ１だけ加算して次のライン番号に相当するベタスクリーン格納先頭アドレスａｄｄｒを取得する。同期用ウエイト（ステップＡ９）は、現在表示に使用されているラインメモリ１６ａまたはラインメモリ１６ｂがまだ使用中であるかどうか、つまり次の水平表示の始まりまで待ってからラインメモリへの書き込みを行うことで、ラインメモリへの２重書き込みを制御している。上記に説明したラインメモリ１６ａまたはラインメモリ１６ｂへの転送をｙ１回行うことで１画素分の表示を行うことが可能となる。

【００４４】

次に複数のウインドウ等を合成して表示する場合について説明する。

ベタスクリーンの表示では、メインメモリ１２にある表示データをその先頭アドレスから順番に読み出して表示するだけであったが、表示プロセッサ２１に与えるプログラムによっては、メインメモリ１２の任意の位置のデータを任意の数だけ取り出したものを任意の組み合せで表示することができる。例えばウインドウシステムの場合、複数のウインドウの表示データをそれぞれ別のアドレスに完成された形でメインメモリ１２上に格納しておき、各ウインドウの位置とプライオリティに従って、表示する際にリアルタイムに重ねあわせて表示することができる。

【００４５】

ここでは図８のメモリマップに示すように、メインメモリ１２上の任意のアドレス位置に背景データ、カーソルデータ、ウインドウ１データ、ウインドウ２偶数データ、ウインドウ２奇数データ等の各種表示データが完成された形で格納されているものとする。これらの表示データのうち、合成したときに表示されるデータのみを読み出してラインメモリに転送する。ウインドウ２偶数データ、ウインドウ２奇数データとは、ＮＴＳＣ信号のようなインタレース信号をメインメモリ１２上に取り込むときに、フィールドごとに偶数データ、奇数データとして取り込んだ場合等のデータ構造である。ただし、カーソルの表示については後述する。

【００４６】

図９は複数のウインドウを合成表示するフローチャートである。図８の各種データを合成したときに表示されるデータのみを読み出して１画面分表示する動作である。

ウインドウ座標やプライオリティ等の表示データはいつ変更されるかわからない任意のデータであるため、１画面を表示するごとに、ステップＢ１にて表示プロセッサ２１によるメインメモリ１２からデータメモリ２０へのデータ転送を行う。また１画面を表示するために、ステップＢ２にて画面Ｙ方向の座標サイズｙ１を取得し、ステップＢ３にて現在表示中の水平ライン番号の次の水平ライン番号をライン番号Ｌとして取得する。αブレンディングするか否かをＢ４にて判断し、αブレンディングしないのであれば通常ライン転送（ステップＢ５）を行い、αブレンディングするのであればαブレンディングライン転送（ステップＢ６）を行う。次にステップＢ７にて、表示するライン番号Ｌと画面Ｙ方向の座標サイズｙ１を比較し、ｙ１回のループが終了していなければラインメモリへの２重書き込みを制御する同期用ウエイト（ステップＢ８）の処理を行い、以上の処理をｙ１回行うことにより１画面分の表示を行う。

【００４７】

図１０はαブレンディングなしの通常ライン転送のフローチャートである。図１１（Ａ）はαブレンディングなしの表示画面例、（Ｂ）はライン番号Ｌのおけ

るラインメモリのメモリマップである。図１１のライン番号Ｌを次に表示されるライン番号Ｌとして、そのライン番号上での通常ライン転送を以下に説明する。ステップＣ１において、表示プロセッサ２１が、αブレンディングなしのライン番号Ｌ上の各表示データどうしの境界ポイントおよびポイント数を算出する。データメモリ２０上に各ウインドウの表示データが転送されて、右上座標、左下座標、Ｘ方向座標サイズ、Ｙ方向座標サイズ、プライオリティ等から、境界ポイントやポイント数が算出される。あるいは、メインＣＰＵ１１によって、あらかじめ算出しておいたデータをデータメモリ２０に転送しておき、データメモリ２０を参照するだけで取得してもよい。

【００４８】

このときの境界ポイントをｘｐｔ［］（［］内は配列順を示す数を記入する）、境界ポイントの数をｘｐｍとして定義する。図１１（Ａ）に示すように、ライン番号Ｌ上の境界ポイントはｘｐｔ［０］＝ｘｓ０，ｘｐｔ［１］＝ｘｓ１，ｘｐｔ［２］＝（ｘｅ１＋１），ｘｐｔ［３］＝（ｘｅ２＋１），ｘｐｔ［４］＝（ｘｅ０＋１）となり、境界ポイント数ｘｐｍは５となる。ステップＣ２にて、境界カウンタｘｐをクリアし、ステップＣ３にてラインＬ上の左境界ポイントｘｐｌを取得し、ステップＣ４にて左境界ポイントの最も近い右境界ポイントｘｐｒを取得する。このｘｐｌ，ｘｐｒ間の表示データを判別し、ステップＣ５にてライン番号Ｌに相当する表示データ格納先頭アドレスａｄｄｒを取得する。最初はｘｐｌ＝ｘｐｔ［０］＝ｘｓ０であり、ｘｐｒ＝ｘｐｔ［１］＝ｘｓ１であることから、この表示データは背景データであることが判別でき、ａｄｄｒ＝ｂａｃｋ＿ａｄｄｒ＋ｘｌ＊Ｌ＋ｘｓ０によりライン番号Ｌに相当する背景データ格納先頭アドレスａｄｄｒが算出できる。

【００４９】

ステップＣ６にてライン番号Ｌが偶数か奇数かを判断し、ラインメモリ１６ａへのデータ転送（ステップＣ７）またはラインメモリ１６ｂへのデータ転送（ステップＣ８）の切り替えを行う。ラインメモリ１６ａおよびラインメモリ１６ｂへのデータ転送サイズは、表示範囲がｘｐｌ，ｘｐｒ－１であるため、ｘｐｒ－ｘｐｌとなる。ラインメモリ１６ａまたはラインメモリ１６ｂへの書き込み位置

はｘｐｌであるので、ラインメモリ１６ａまたはラインメモリ１６ｂへのデータ転送はａｄｄｒから（ｘｓ１－ｘｓ０）のデータをラインメモリ１６ａまたはラインメモリ１６ｂのｘｓ０に転送することになる。右境界ポイントｘｐｒは次のｘｓ１，（ｘｅ１＋１）間のデータを転送するときには左境界ポイントｘｐｌとなるため、ステップＣ９にてｘｐｌ＝ｘｐｒとすることで左境界ポイントｘｐｌを取得することができる。既に説明した右境界ポイントｘｐｒの取得（ステップＣ４）に移行し、これらの動作をｘｓ１，（ｘｅ１＋１）、（ｘｅ１＋１），（ｘｅ２＋１）、（ｘｅ２＋１），（ｘｅ０＋１）の境界間でも同様に行うことにより、ライン番号Ｌの１ラインのデータ転送を行うことができる。ステップＣ１０にて境界カウントｘｐと境界ポイント数ｘｐｍの比較を行い、境界カウントｘｐが境界ポイント数ｘｐｍと同じか大きくなることにより次ラインの処理へと移行する。

【００５０】

図１２はαブレンディングを含んだライン転送のフローチャートである。図１３はαブレンディングを含む表示画面例である。図１３（Ａ）はαブレンディングの表示画面例、（Ｂ）はライン番号Ｌのおける通常ラインメモリとαブレンディング用ラインメモリのメモリマップである。図１３のライン番号Ｌを次に表示されるライン番号Ｌとして、そのライン番号上でのαブレンディングライン転送を以下に説明する。ステップＤ１において、表示プロセッサ２１は、αブレンディングのあるライン番号Ｌ上での各表示データどうしの境界ポイントおよびポイント数を算出する。図１１の通常表示画面例よりも境界ポイント数が１つ増えている。この境界ポイントやポイント数はデータメモリ２０上に転送されて得られた各表示データの右上座標、左下座標、Ｘ方向座標サイズ、Ｙ方向座標サイズ、プライオリティ等により算出される。あるいは、メインＣＰＵ１１によってあらかじめ算出しておいたデータをデータメモリ２０に転送しておき、データメモリ２０を参照するだけで取得してもよい。

【００５１】

ライン番号Ｌ上の境界ポイントはｘｐｔ［０］＝ｘｓ０，ｘｐｔ［１］＝ｘｓ１，ｘｐｔ［２］＝ｘｓ２，ｘｐｔ［３］＝（ｘｅ１＋１），ｘｐｔ［４］＝（

ｘｅ２＋１），ｘｐｔ［５］＝（ｘｅ０＋１）となり境界ポイント数ｘｐｍは６となる。αブレンディングのない境界は通常ライン転送と同じであるため、αブレンディングのある境界カウンタｘｐのときについて説明する。ステップＤ１４によって取得されるラインＬ上の左境界ポイントｘｐｌはｘｐｌ＝ｘｐｒ＝ｘｐｔ［２］＝ｘｓ２であり、Ｄ４による右境界ポイントｘｐｒの取得によりｘｐｒ＝ｘｐｔ［３］＝（ｘｅ１＋１）となる。この表示データのライン番号Ｌに相当するウインドウ１データ格納先頭アドレスａｄｄｒは、ａｄｄｒ＝ｗｉｎ１＿ａｄｄｒ＋（ｘｅ１－ｘｓ１＋１）＊（Ｌ－ｙｓ１）＋（ｘｓ２－ｘｓ１）と算出される（ステップＤ５）。ライン番号Ｌが偶数か奇数かをステップＤ６にて判断し、ラインメモリ１６ａへのデータ転送（ステップＤ７）またはラインメモリ１６ｂへのデータ転送（ステップＤ８）の切り替えを行う。

【００５２】

ラインメモリ１６ａまたはラインメモリ１６ｂへのデータ転送サイズは、表示範囲がｘｐｌ，ｘｐｒ－１であるため、ｘｐｒ－ｘｐｌとなる。ラインメモリ１６ａまたはラインメモリ１６ｂへの書き込み位置はｘｐｌであるから、ラインメモリ１６ａまたはラインメモリ１６ｂへのデータ転送はａｄｄｒから（（ｘｅ１＋１）－ｘｓ２）のデータをラインメモリ１６ａまたはラインメモリ１６ｂのｘｓ２に転送することになる。データ転送終了後、そのデータに対してαブレンディングする別データがあるか否かをステップＤ９にて判断する。ここではウインドウ１とウインドウ２がαブレンディングであり、この表示データのライン番号Ｌに相当するウインドウ２偶数データ格納先頭アドレスａｄｄｒは、ａｄｄｒ＝ｗｉｎ２ｅ＿ａｄｄｒ＋（ｘｅ２－ｘｓ２＋１）＊（Ｌ－ｙｓ２）と算出される（ステップＤ１０）。

【００５３】

ライン番号Ｌが偶数か奇数かをステップＤ１１で判断してラインメモリ１６ｃへのデータ転送（ステップＤ１２）またはラインメモリ１６ｄへのデータ転送（ステップＤ１３）の切り替えを行う。このときのラインメモリ１６ｃまたはラインメモリ１６ｄはαブレンディング用ラインメモリである。ラインメモリ１６ｃまたはラインメモリ１６ｄへのデータ転送サイズは、表示範囲がｘｐｌ，ｘｐｒ

－１であるため、ｘｐｒ－ｘｐｌとなる。ラインメモリ１６ｃまたはラインメモリ１６ｄへの書き込み位置はｘｐｌであるから、ラインメモリ１６ｃまたはラインメモリ１６ｄへのデータ転送は、ａｄｄｒから（（ｘｅｌ＋１）－ｘｓ２）のデータをラインメモリ１６ｃまたはラインメモリ１６ｄのｘｓ２に転送することになる。通常ラインメモリにはαブレンディングしないデータを、αブレンディング用ラインメモリにはαブレンディングするデータを別々に持つことができ、ハードウエアのαブレンディング処理により合成表示を行うことが可能となる。以降の処理であるステップＤ１４、ステップＤ１５は通常ライン転送と同様である。

【００５４】

カーソルの表示は上記に記載した動作手順によっても表示することができるが、上記の１ライン分の表示データをラインメモリに転送した後に、カーソルの座標、カーソルＸ方向サイズ、カーソルＹ方向のサイズ、カーソルデータ格納先頭アドレスｃｕｒｓ＿ａｄｄｒ等を与えて、最後に合成表示させることによっても実現できる。αブレンディングのデータ上に表示する場合は通常のラインメモリとαブレンディング用のラインメモリの両方に書き込むことで、カーソルの表示が行える。この方法ではカーソルは常に最上位の優先順位となり、処理速度を早くすることができる。以上が表示プロセッサ２１の基本動作の説明である。

【００５５】

次に表示プロセッサ２１が行う他の動作について説明する。

まず表示データの拡大・縮小・スキップの処理について説明する。図３に示したように、表示プロセッサ２１は内部に転送用バッファメモリを２組持っている。メインメモリ１２から読み込まれた表示データは、まず１組目の転送用バッファメモリ２５ａ，２５ｂに格納され、次にもう１組の転送用バッファメモリ２６ａ，２６ｂに格納されたのちに、表示用のラインメモリ１６へ格納される。この転送用バッファメモリ間の読み出しおよび書き込みは、表示プロセッサ２１に与えるプログラムによって細かく制御できる。

【００５６】

具体的には、１組目の転送用バッファメモリ２５ａ，２５ｂ（読み出しメモリ

と呼ぶ）の読み出しカウンタのスタート／ストップ、もう１組の転送用バッファメモリ２６ａ，２６ｂ（書き込みメモリと呼ぶ）への書き込みカウンタのスタート／ストップ及び書き込みのする／しないを画素単位で任意の位置で行うことができる。これにより表示画像の拡大、縮小や、ある位置より右側の画像が右方向にずれて画像に穴が開いたように見える表現（スキップと呼ぶ）およびそれらを混在させた表示データに変化させることができる。

【００５７】

拡大・縮小・スキップの動作は図１４に示すコントロールデータの動作説明図に示すように制御される。コントロールデータは１画素につき２ビットの情報を持ち、画素単位で転送用バッファメモリ２５ａ，２５ｂ，２６ａ，２６ｂ間の読み出しカウンタおよび書き込みカウンタおよび書き込みのする／しないを制御する。図１５は拡大・縮小・スキップを行わない等倍のときの転送用バッファメモリ間の転送動作であり、この場合、コントロールデータとして“００”を与え続ける。すると、読み出しカウンタ・書き込みカウンタとも１ずつカウントアップされていき、読み出しメモリと同じデータが書き込みメモリに書き込まれて等倍の転送となる。

【００５８】

縮小を行う場合は、コントロールデータの省きたい画素に対応するデータを“０１”にする。縮小動作を示す図１６において、書き込みメモリには、０、１、２、３までは順に表示データが書き込まれるが、３の位置のコントロールデータが“０１”であるため、書き込みカウンタがストップし、次に３の位置に４を重ね書きする。これで表示データが１画素分だけ縮小する。コントロールデータに１画素おきに“０１”を設定すれば画像の水平方向は１／２に縮小するし、部分的に“０１”を設定する割合を変えれば、例えば画像が円柱形になったりする。

【００５９】

拡大を行う場合には、コントロールデータの対応する位置に“１０”を設定する。図１７において、書き込みメモリには、０、１、２、３までは順に表示データが書き込まれるが、３の位置のコントロールデータが“１０”であるため、読み出しカウンタがストップし、次に３の隣にもう一度３が書かれる。これで１画

素分の拡大が行われる。

【００６０】

コントロールデータが“１１”のときはスキップである。図１８で０、１、２まではそのまま書かれるが、３の位置のコントロールデータが“１１”であるため、読み出しアドレスが停止する。このため３の表示データは右隣りの画素に書かれることになる。さらに書き込みメモリへの書き込みが行われず、書き込みメモリの３の位置には何も書かれない。これで１画素分のスキップが行われる。

【００６１】

以上のようにコントロールデータの値を設定することにより、拡大・縮小・スキップが可能であり、また図１９～図２１のように拡大・縮小・スキップを混在して設定することにより一部は拡大するが他の一部は縮小するといったような複雑な表示データの変形が行える。

【００６２】

ところで拡大・縮小率が水平方向に一定であることは多いが、この場合コントロールデータは同じパターンの繰り返しとなる。本実施形態では繰り返すパターンと繰り返しポイントを設定することにより、１水平ライン分のコントロールデータを書き込むのに比べ、少ないデータで拡大・縮小等の指定をすることができる。例えば０．７５倍に縮小する場合には図２２のようにコントロールデータは“００”、“００”、“００”、“０１”の繰り返しとなる。この場合、この４画素分のコントロールデータと４画素単位で繰り返しが行われるように繰り返しポイントの設定をすることにより、同じコントロールデータが繰り返し使用され、縮小動作が行われる。同様に図２３は１．７５倍に拡大する場合である。

【００６３】

次に、本実施形態ではビデオ入力を２系統持っているが、表示プロセッサ２１は、これよりビデオ映像データの取り込みを行うことができる。ビデオ映像信号はＡ／Ｄ変換された後、ビデオ入力用ラインメモリに格納される。ビデオ入力用ラインメモリはビデオ入力１系統につき２本あり、他のラインメモリと同様に読み出しと書き込みを行うメモリを交互に切り替えて使用する。ビデオ入力用ラインメモリに書き込まれたビデオデータは、表示用プロセッサ２１によって読み出

され、表示プロセッサ２１内で拡大・縮小・スキップ処理などを行った後、ラインメモリ１６に転送される。

【００６４】

次にデータ処理回路１３について説明する。メインメモリ１２に格納されている表示データは通常のＲＧＢ形式のデータだけでなく、さまざまなデータ形式で格納されている。表示プロセッサ２１によってメインメモリ１２ら表示データが読み出されラインメモリ１６に書き込まれる間に、ＹＵＶデコーダ２７ａ、ランレングス展開回路２７ｂ、カラー伸長回路２７ｃ、カラーパレット２７ｄ，２７ｅの処理回路があり、そこで各種データ形式の表示データはＲＧＢ形式に変換されてラインメモリ１６に格納される。どのデータ処理回路によって変換を行うかは、表示プロセッサ２１が画素単位にセレクタ２８に指示して選択させる。カラーパレットは複数持つことができ、例えばウインドウごとに使うパレットを変えることができる。また、さらに他のデータ処理回路を追加
することで、さまざまな表示データのフォーマットに対応することができる。

【００６５】

データ処理回路１３を通過した表示データはラインメモリ１６に書き込まれるが、表示データのうちいくつかの値を、実際には表示されないライトスルーデータとして設定することができる。メインメモリ１２やデータバッファ１５から表示プロセッサ２１が表示データをラインメモリ１６に転送する際、ライトスルーデータがあると、その画素についてはラインメモリ１６への書き込みを行わない。これは矩形でない画像、例えばマウスカーソルなどの表示に有効である。

【００６６】

次に使用ライン情報を用いて画面表示を行う動作について説明する。通常、表示用のラインメモリは２本一組で動作する。これは表示のために読み出しを行っているラインメモリに対して、表示プロセッサ２１が書き込みアクセスを行うことができないためで、読み出しを行っているラインメモリとは別のもう一方のラインメモリに次のラインの表示データの書き込みを行う。表示するラインが変わるたびに、この読み込みと書き込みを行うラインメモリを交互に入れ替えて表示を進めていく。ところが図４（Ｂ）、（Ｃ）のように複数の画面を合成して表示

する場合で、特に背景を表示しないときには、ラインメモリへの表示データの書き込みは、ウインドウを表示する部分についてのみ行われ、その他の部分には前のラインの表示データが残ったままになることがある。そのため書き込み前にラインメモリのクリアが必要となり、そのための時間が必要になってくる。使用ライン情報は、このラインメモリのクリア作業を不要にするものである。

【００６７】

使用ライン情報は、ラインメモリ上の各画素の表示データに１対１で対応し、その表示データが何ライン目の表示で使用されるデータであるかを表す情報である。表示データ１画素に対応する使用ライン情報は（画面の垂直方向の画素数＋１）を表現できるビット数（画面サイズが１２８０×１０２４ならば１１ビット）以上で、それが各ラインメモリに表示データと同じ画素数分、つまり水平画素数分だけである。

【００６８】

図２４は、使用ライン情報を格納する表示メモリ部１４を示すブロック図である。ラインメモリ１６ａ～１６ｆにはそれぞれコンパレータ３１～３６、ＡＮＤ回路３７～４２が接続されている。ラインメモリ１６ｅ，１６ｆは後述する背景データを格納するメモリである。コンパレータ３１～３６は、表示ライン数と使用ライン情報を比較し、値が一致する場合は論理値１を出力し、不一致の場合は論理値０を出力する。ＡＮＤ回路３７～４２は、論理値１が入力されると、表示データをそのまま出力し、論理値０が入力されると表示データを出力しない。

【００６９】

以下、画面表示動作について図２５をもとに説明する。図２５（Ａ）は表示画面例、（Ｂ）は使用ライン情報がＮの場合のラインメモリのメモリマップと出力データ、（Ｃ）は使用ライン情報がＮ＋２の場合のラインメモリのメモリマップと出力データ、（Ｄ）は使用ライン情報がＮ＋４の場合のラインメモリのメモリマップと出力データである。同図（Ｂ）に示すように、（Ｎ－１）ライン目の表示を行っている間、表示プロセッサ２１はラインメモリにＮライン目の表示データを書き込む。Ｎライン目にはウインドウ１があり、ウインドウ１の表示データ６書き込む際に同時に使用ライン情報にＮを書き込む。Ｎライン目を表示する際

には、ラインメモリの１画素毎に、表示中のライン番号Ｎと使用ライン情報との比較を行い、それが等しい場合のみ表示データ有効とみなし、ラインメモリ中の表示データを出力する。

【００７０】

次に同じラインメモリに書き込みを行うのは、ラインメモリを２本交互に使用するため（Ｎ＋２）ライン目である。同図（Ｃ）に示すように、（Ｎ＋２）ライン目には、ウインドウ１とウインドウ２の２つがあり、その表示データと使用ライン情報には（Ｎ＋２）を書き込む。これを同様にして表示を行う。

【００７１】

次に（Ｎ＋４）ライン目の書き込みを行う。（Ｎ＋４）ライン目はウインドウ２についてのみであり、同図（Ｄ）に示すように、表示データと使用ライン情報に（Ｎ＋４）の書き込みを行う。このとき（Ｎ＋２）ライン目で書き込んだウインドウ１のデータが残ったままになっており、何らかの工夫を行わない場合はこれが表示されてしまい、間違った表示となる。ところが本実施形態では、この古いウインドウ１の部分の使用ライン情報は（Ｎ＋２）のままなので無視されて、ウインドウ２のみ正しく表示される。

【００７２】

このようにしてすべてのラインについて表示を行うが、垂直帰線期間毎にすべてのラインメモリの使用ライン情報をクリアする必要がある。これは前の垂直表示期間の表示データが表示されるのを防ぐためである。なおクリアは、使用ライン情報として使用されていない値を書き込むことによって行う。

【００７３】

次に同じパターンの繰り返し表示について説明する。ウインドウシステムの背景画面などによく見られるが、水平方向に同じパターンが繰り返し表示されることがよくある。この場合ラインメモリ１６から読み出す読みだしアドレスを任意の範囲でループできる様にすることで、特定のパターンを繰り返し表示できる。それにより特に背景データをメインメモリ１２に格納している場合などは、読み出して来るデータ量を削減でき、メインＣＰＵ１１のデータバスのトラフィックを低下させることができる。この機能を使用する際には、通常のラインメモリの

他に繰り返しパターンを格納する専用のラインメモリ１６ｅ，１６ｆが２本一組必要である。よってラインメモリは最低で４本、αブレンディングを同時に使用する場合は最低６本必要になる。この特定パターンの繰り返し表示機能について以下にに説明する。

【００７４】

図２６は、背景を繰り返し利用する場合の動作説明図である。Ｎライン目についてラインメモリの書き込みを行う場合、まず通常と同様にウインドウデータを収納するラインメモリにウインドウの表示データおよび使用ライン情報Ｎを書き込む。次に背景データを収納するラインメモリに背景の表示データと使用ライン情報Ｎを書き込み、さらに繰り返しポイントを設定する。繰り返しポイントの設定方法にはいくつかの方法が考えられ、専用のレジスタを設けたり、使用ライン情報や表示データに通常と区別できる値を書き込んだり、専用のラインメモリを用意するといった方法が考えられる。

【００７５】

表示の際には、まずウインドウデータを収納するラインメモリの使用ライン情報を表示中のライン番号と比較する。一致すればウインドウの表示データを出力し、一致しない場合は背景データを出力する。背景データは図示していないが内部の背景データ読み出しカウンタによって示される背景データが出力される。この読み出しカウンタの値が繰り返しポイントの値と一致したならば、読み出しカウンタの値がクリアされる。これにより出力される背景データは、背景データを収納するラインメモリの最初に戻り、背景データがこの範囲で繰り返して出力される。

【００７６】

次にデータバッファ１５について説明する。表示データはメインメモリ１２上に格納するのが通常であるが、カーソルなどサイズが小さくパターンが決まっている表示データはデータバッファ１５に格納するとよい。データバッファ１５に格納された表示データは、表示プロセッサ２１によってラインメモリ１６に書き込むことができる。またラインメモリ１６ではなく表示プロセッサ２１のプログラムメモリ１９やデータメモリ２０やメインメモリ１２に転送することもできる

ので、カーソルの表示など限らず汎用に使用することができる。

【0077】

また、αブレンディングで2画面の混合比率を設定する方法にはいくつか考えられる。ひとつは混合比率を格納する専用レジスタを用意して、αブレンディングするときにそのレジスタより混合比率を読み出す方法がある。その場合、混合比率が変化するたびに表示プロセッサ21がレジスタの内容を書き換える必要がある。他には混合比率を複数格納するＬＵＴを用意し、ラインメモリに表示データを書き込む際に、そのＬＵＴの呼び出しアドレスを一緒に画素単位でラインメモリに書き込む方法、または直接混合比率をラインメモリに画素単位で書き込んでしまう方法などが考えられる。

【0078】

【発明の効果】

請求項1の発明によれば、表示をする際に必要な部分の表示データをメインメモリ内から取り出して使用するため、メインメモリ内の任意の位置のデータを取り出して組み合わすことが可能である。この制御はすべて表示制御部が行うので、主制御部が表示のためにスクリーン上に複数のウインドウを同時表示する等の際のソフトウエアにおける処理負荷を低減でき、各ウインドウの移動や切り替えを高速化できる。

【0079】

請求項2の発明によれば、ラインメモリ上のデータを読み出す際、それがライン方向に対して繰り返すようなデータであった場合（ウインドウシステムにおける背景等）、読み出しラインメモリアドレスを任意の位置でループできるため、冗長な処理が不要になり、処理の高速化が図れる。

【0080】

請求項3の発明によれば、カーソルや繰り返し背景などをデータバッファメモリに収納しておけるため、決まりきったデータをメインメモリから読み出す必要が無いため、データバスの負荷を減らし、冗長な処理が不要になり、処理の高速化が図れる。

【００８１】

請求項４の発明によれば、表示データを読み出す際に拡大縮小処理をするため、表示用データに対する拡大縮小処理を事前にする必要が無く、バスの使用効率を上げられる。また、ビデオ入力映像を表示する場合に映像サイズの変更が必要となるのが常であるが、出力段に拡大縮小処理を掛けることで拡大縮小回路がより有効に利用できる。また、この事によりビデオデータを常にフルサイズで取り込みながら、そのデータを一旦フレームメモリなどに転送することなく表示は任意のサイズが行える。

【００８２】

請求項５の発明によれば、第１バッファメモリからの読みだしアドレスカウントを所定の順に停止／動作を繰り返すことにより、一定倍率の拡大・縮小が簡単な処理で行うことができ、処理の高速化ができる。

【００８３】

請求項６の発明によれば、表示制御部は、格納情報のデータ形式情報に基づいてデータ変換ができるので、表示用データを収納する形式などに制限が無いため、データメモリ上に収納されている表示キャラクタ等をわざわざフレームバッファ等に転送する必要が無く、処理の高速化が図れる。

【００８４】

請求項７の発明によれば、前記表示制御部に必要なプログラムとデータを格納するプログラムメモリとデータメモリとを備えるので、メインメモリから処理の度にメインメモリからデータを読み出す必要がなく、データバスの使用回数を減らし、処理の高速化を図ることができる。

【００８５】

請求項８の発明によれば、前記表示制御部は、前記プログラムメモリとデータメモリに必要な情報をメインメモリから転送させるため、画面モードあるいはグラフィック領域の変更に柔軟に対応できる。容量を越えたプログラムあるいはデータは、メインメモリから読み出せばよいので、小容量で済み、コンパクトあるいは低コストでシステムを構築できる。

【００８６】

請求項９の発明によれば、各ラインメモリに表示データを転送する際、そのデータを使用するライン番号を同時に、１ドット毎に対応した使用ライン情報メモリに書込み、表示する際に表示しようとしているラインの番号と比較することによりラインメモリ上のデータが有効であるかどうかを判別することで、ラインメモリを使用する前に毎回ラインメモリの内容をクリアする必要が無くなり、処理の高速化が図れる。各ラインの表示毎にラインメモリ内のデータを消去する必要がなく、垂直帰線期間毎にすべてのラインメモリの使用ライン情報を消去するだけでよいので処理の高速化を図れる。

【図面の簡単な説明】

【図１】

本発明に係る画面表示装置の一実施形態を示すブロック図である。

【図２】

この画面表示装置のデータ処理回路と表示メモリ部を示すブロック図である。

【図３】

この画面表示装置の表示プロセッサを示すブロック図である。

【図４】

（Ａ）～（Ｃ）は、メインメモリの表示データとディスプレイの表示出力を示す説明図である。

【図５】

ベタスクリーンデータを１画面分表示するフローチャートである。

【図６】

ベタスクリーンの表示画面例である。

【図７】

ベタスクリーンデータが格納されているメインメモリのメモリマップである。

【図８】

各種表示データが格納されているメインメモリのメモリマップである。

【図９】

複数のウインドウを合成表示するフローチャートである。

【図１０】

αブレンディングなしの通常ライン転送のフローチャートである。

【図１１】

（Ａ）はαブレンディングなしの表示画面例、（Ｂ）はライン番号Ｌのおけるラインメモリのメモリマップである。

【図１２】

αブレンディングを含んだライン転送のフローチャートである。

【図１３】

（Ａ）はαブレンディングの表示画面例、（Ｂ）はライン番号Ｌのおける通常ラインメモリとαブレンディング用ラインメモリのメモリマップである。

【図１４】

コントロールデータの動作内容を示す説明図である。

【図１５】

拡大・縮小・スキップを行わない等倍のときの転送用バッファメモリ間の転送動作の説明図である。

【図１６】

転送用バッファメモリの縮小動作を示す説明図である。

【図１７】

転送用バッファメモリの拡大動作を示す説明図である。

【図１８】

転送用バッファメモリのスキップ動作を示す説明図である。

【図１９】

拡大・縮小・スキップの混在した転送用バッファメモリの動作を示す説明図である。

【図２０】

拡大・縮小・スキップの混在した転送用バッファメモリの他の動作を示す説明図である。

【図２１】

拡大・縮小・スキップの混在した転送用バッファメモリの更に他の動作を示す

説明図である。

【図２２】

転送用バッファメモリの一定倍率の縮小動作を示す説明図である。

【図２３】

転送用バッファメモリの一定倍率の拡大動作を示す説明図である。

【図２４】

使用ライン情報を格納する表示メモリ部を示すブロック図である。

【図２５】

（Ａ）は表示画面例、（Ｂ）は使用ライン情報がＮの場合のラインメモリのメモリマップと出力データ、（Ｃ）は使用ライン情報がＮ＋２の場合のラインメモリのメモリマップと出力データ、（Ｄ）は使用ライン情報がＮ＋４の場合のラインメモリのメモリマップと出力データである。

【図２６】

背景を繰り返し利用する場合の動作説明図である。

【図２７】

従来の画像表示装置の一例を示すブロック図である。

【符号の説明】

１１　メインＣＰＵ

１２　メインメモリ

１３　データ処理回路

１４　表示メモリ部

１５　データバッファ

１６　ラインメモリ

１７　出力処理回路

１８　ＤＭＡ

１９　プログラムメモリ

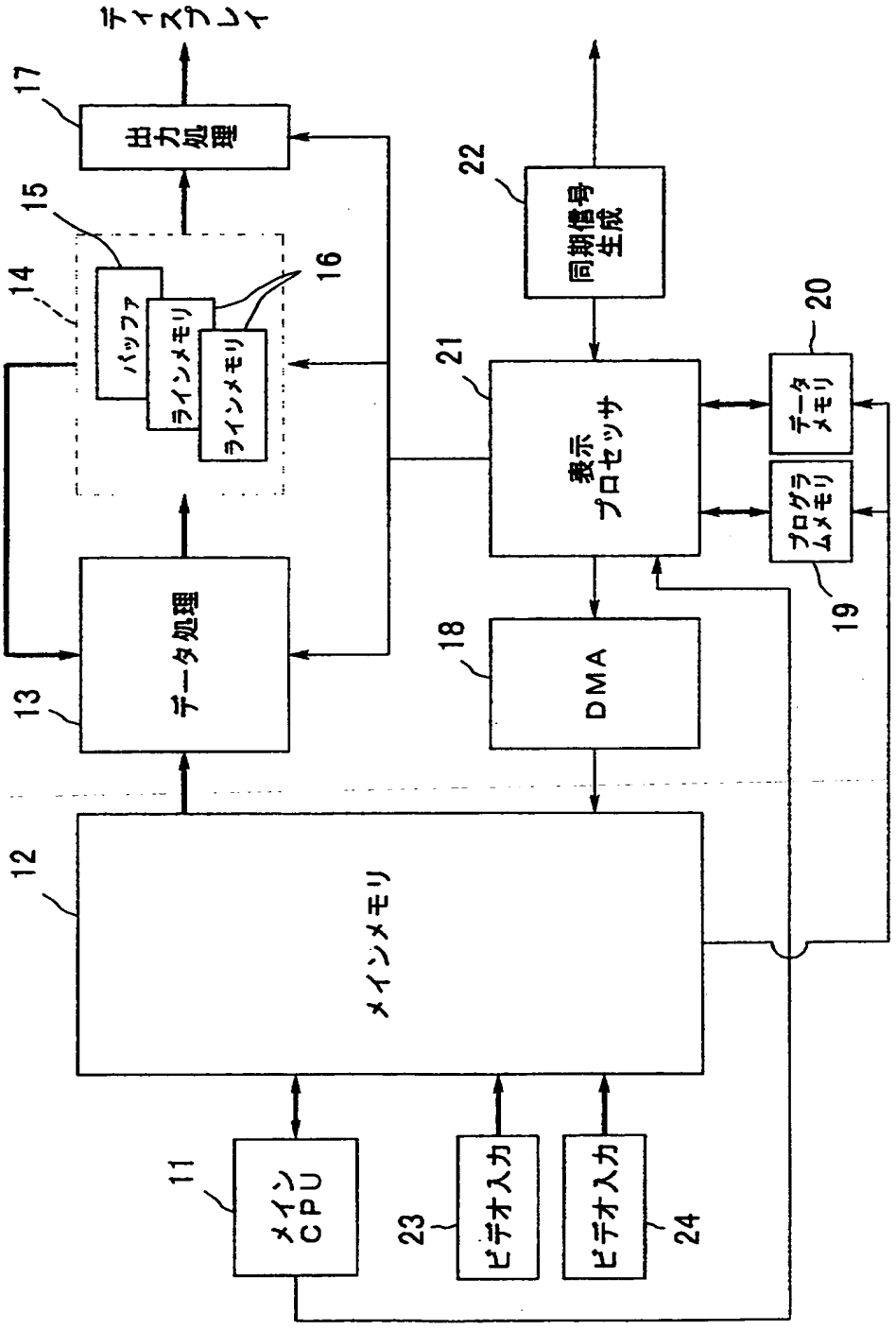
２０　データメモリ

２１　表示プロセッサ

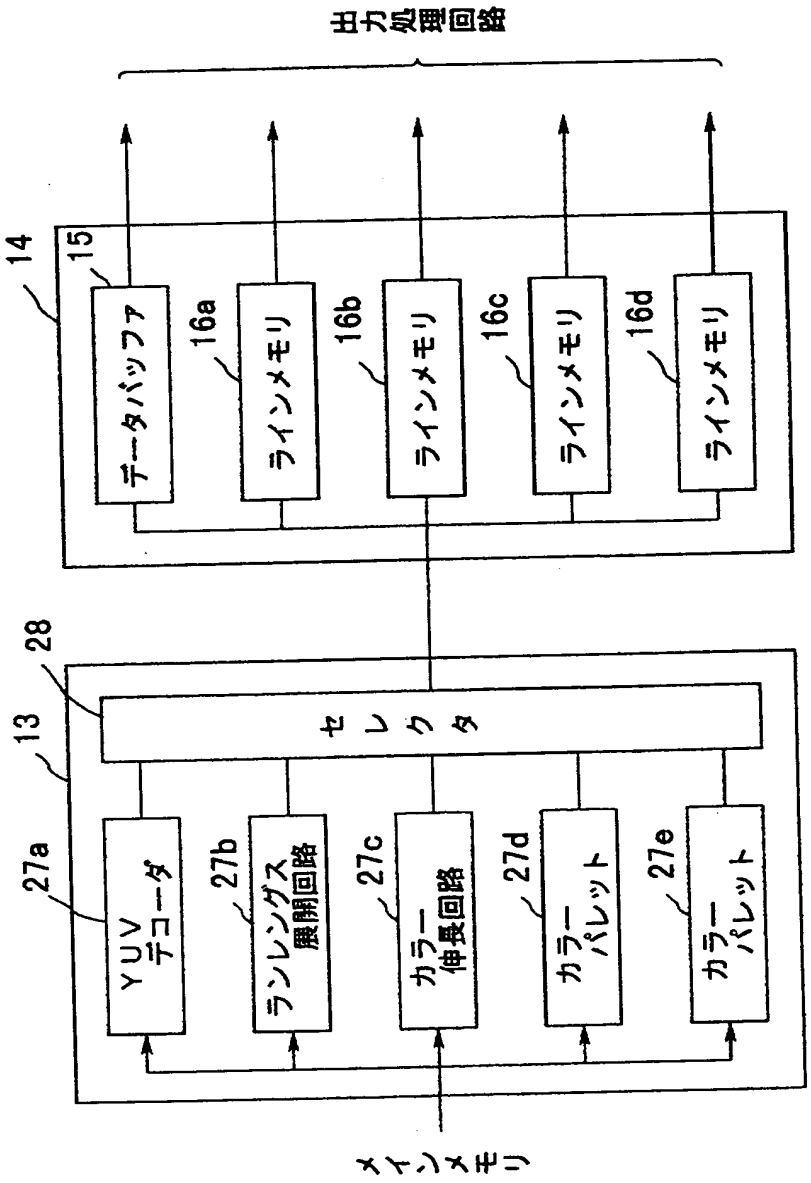
２２　同期信号生成回路

【書類名】　図面

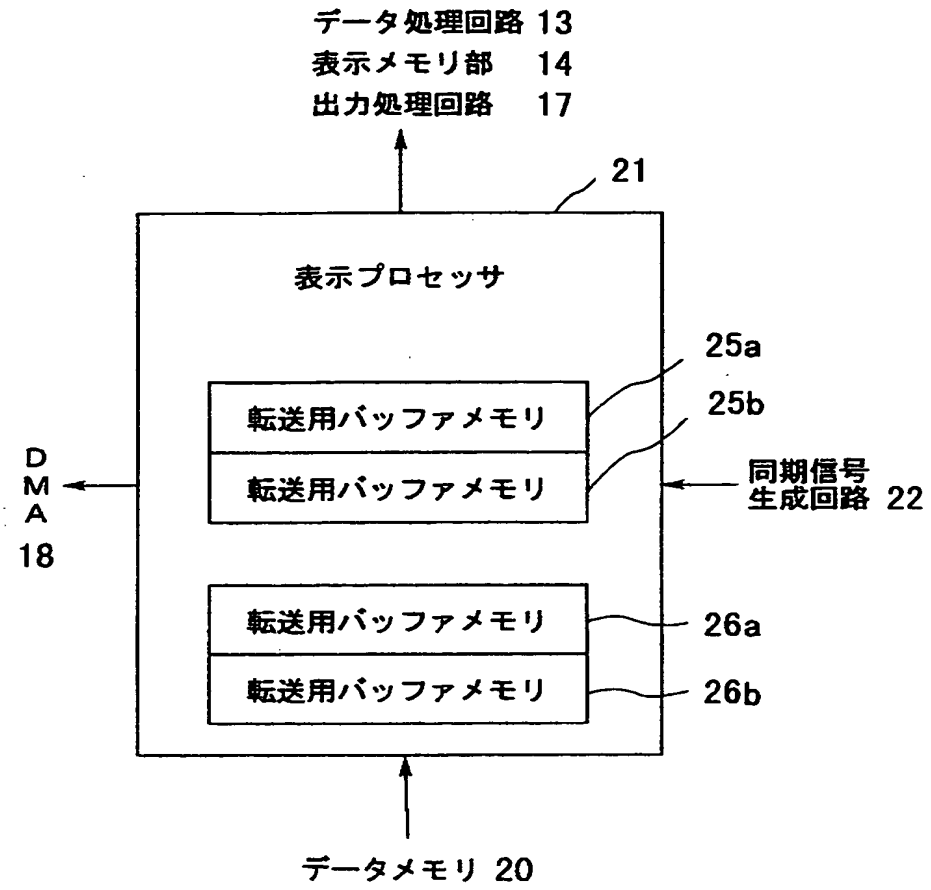
【図１】


ディスプレイ
出力処理
17
15
14
16
バッファ
ラインメモリ
ラインメモリ
22
同期信号生成
20
21
表示プロセッサ
データメモリ
プログラムメモリ
19
データ処理
13
18
DMA
12
メインメモリ
11
メインCPU
23
ビデオ入力
24
ビデオ入力

【図２】

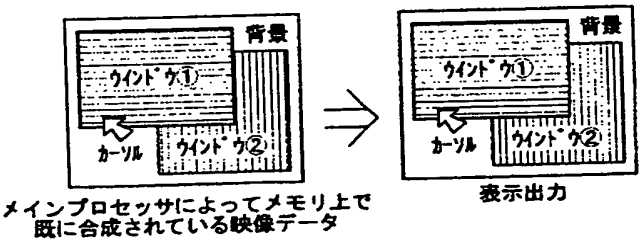

出力処理回路
14
15
データバッファ
16a
ラインメモリ
16b
ラインメモリ
16c
ラインメモリ
16d
ラインメモリ
28
セレクタ
13
27a
YUV デコーダ
27b
ランレングス 展開回路
27c
カラー 伸長回路
27d
カラー パレット
27e
カラー パレット
メインメモリ

【図３】

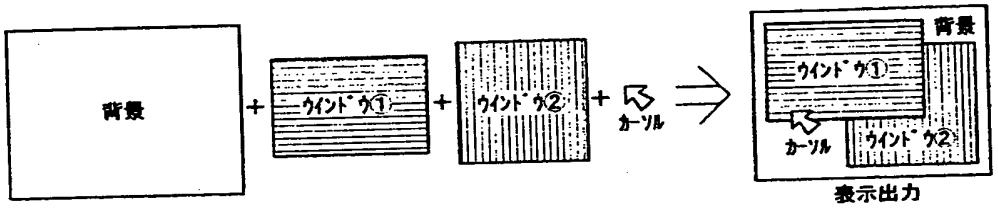

データ処理回路 13
表示メモリ部　14
出力処理回路　17
21
表示プロセッサ
25a
25b
転送用バッファメモリ
転送用バッファメモリ
DMA
18
同期信号
生成回路 22
転送用バッファメモリ
転送用バッファメモリ
26a
26b
データメモリ 20

【図４】

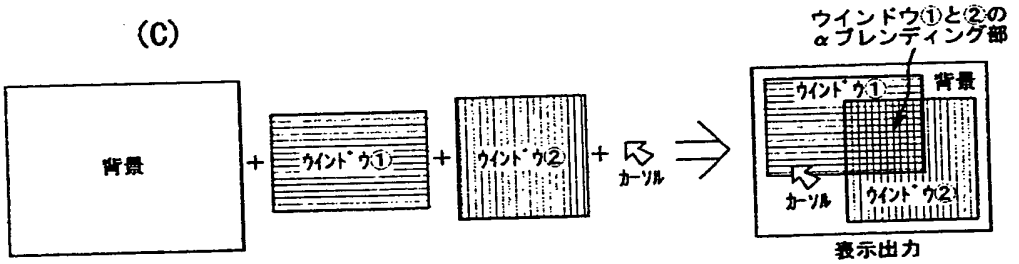
(A)


背景
ｳｲﾝﾄﾞｳ①
ｶｰｿﾙ
ｳｲﾝﾄﾞｳ②
背景
ｳｲﾝﾄﾞｳ①
ｶｰｿﾙ
ｳｲﾝﾄﾞｳ②
メインプロセッサによってメモリ上で
既に合成されている映像データ
表示出力

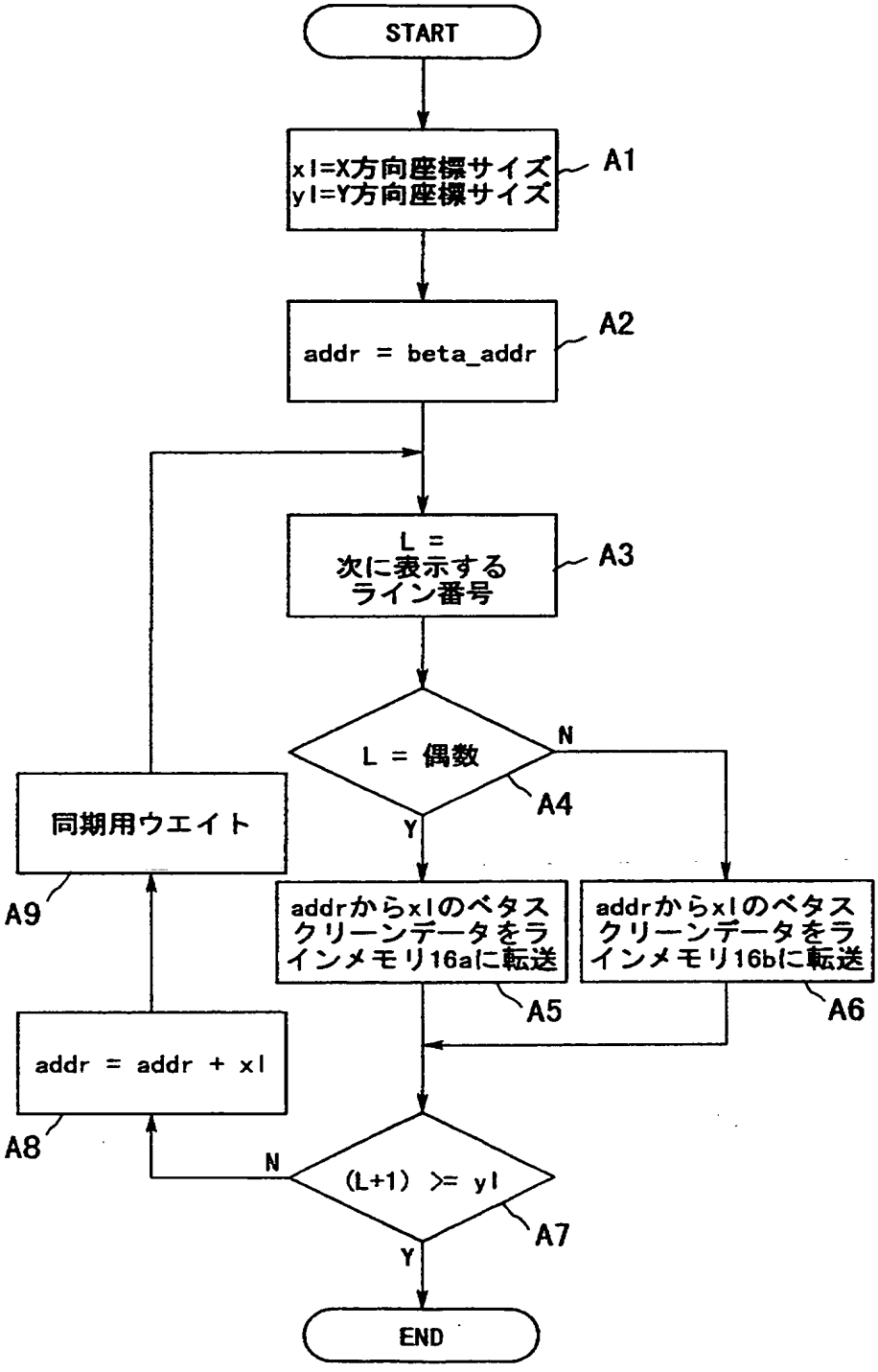
(B)


背景
ｳｲﾝﾄﾞｳ①
ｳｲﾝﾄﾞｳ②
ｶｰｿﾙ
背景
ｳｲﾝﾄﾞｳ①
ｶｰｿﾙ
ｳｲﾝﾄﾞｳ②
表示出力

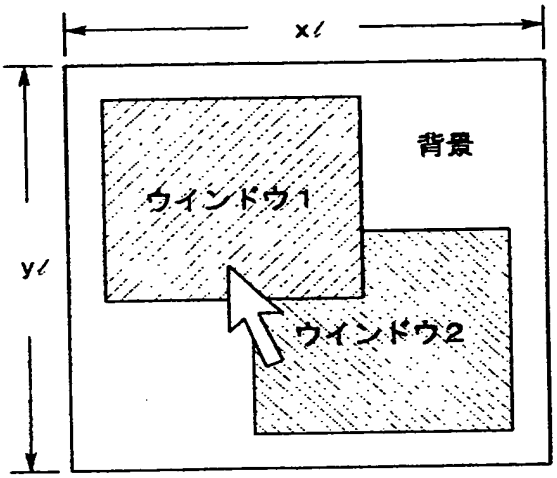
(C)


ウインドウ①と②の
αブレンディング部
背景
ｳｲﾝﾄﾞｳ①
ｳｲﾝﾄﾞｳ②
ｶｰｿﾙ
ｳｲﾝﾄﾞｳ①
背景
ｶｰｿﾙ
ｳｲﾝﾄﾞｳ②
表示出力

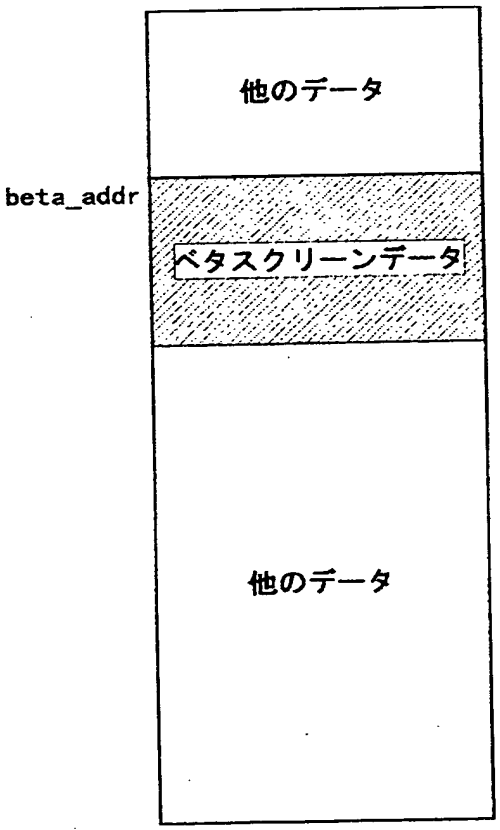
【図５】


START
xl=X方向座標サイズ
yl=Y方向座標サイズ
A1
addr = beta_addr
A2
L =
次に表示する
ライン番号
A3
L = 偶数
N
Y
A4
同期用ウエイト
A9
addrからxlのベタ
クリーンデータをラ
インメモリ16aに転送
A5
addrからxlのベタ
クリーンデータをラ
インメモリ16bに転送
A6
addr = addr + xl
A8
N
(L+1) >= yl
A7
Y
END

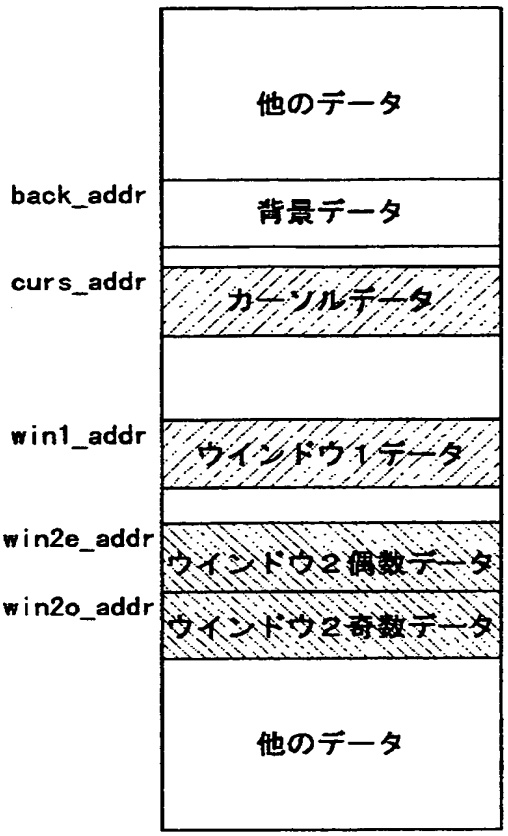
【図６】



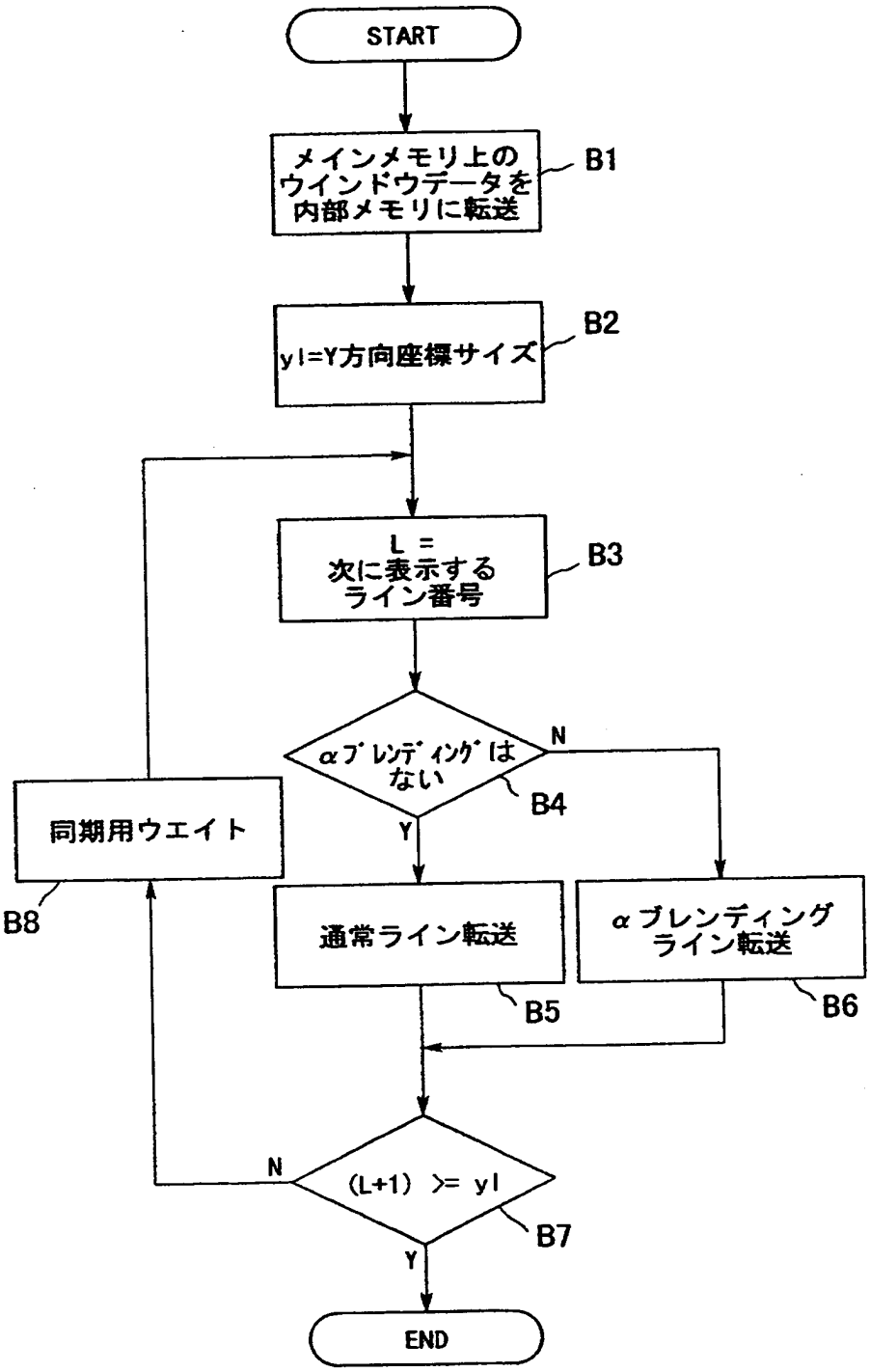
【図７】



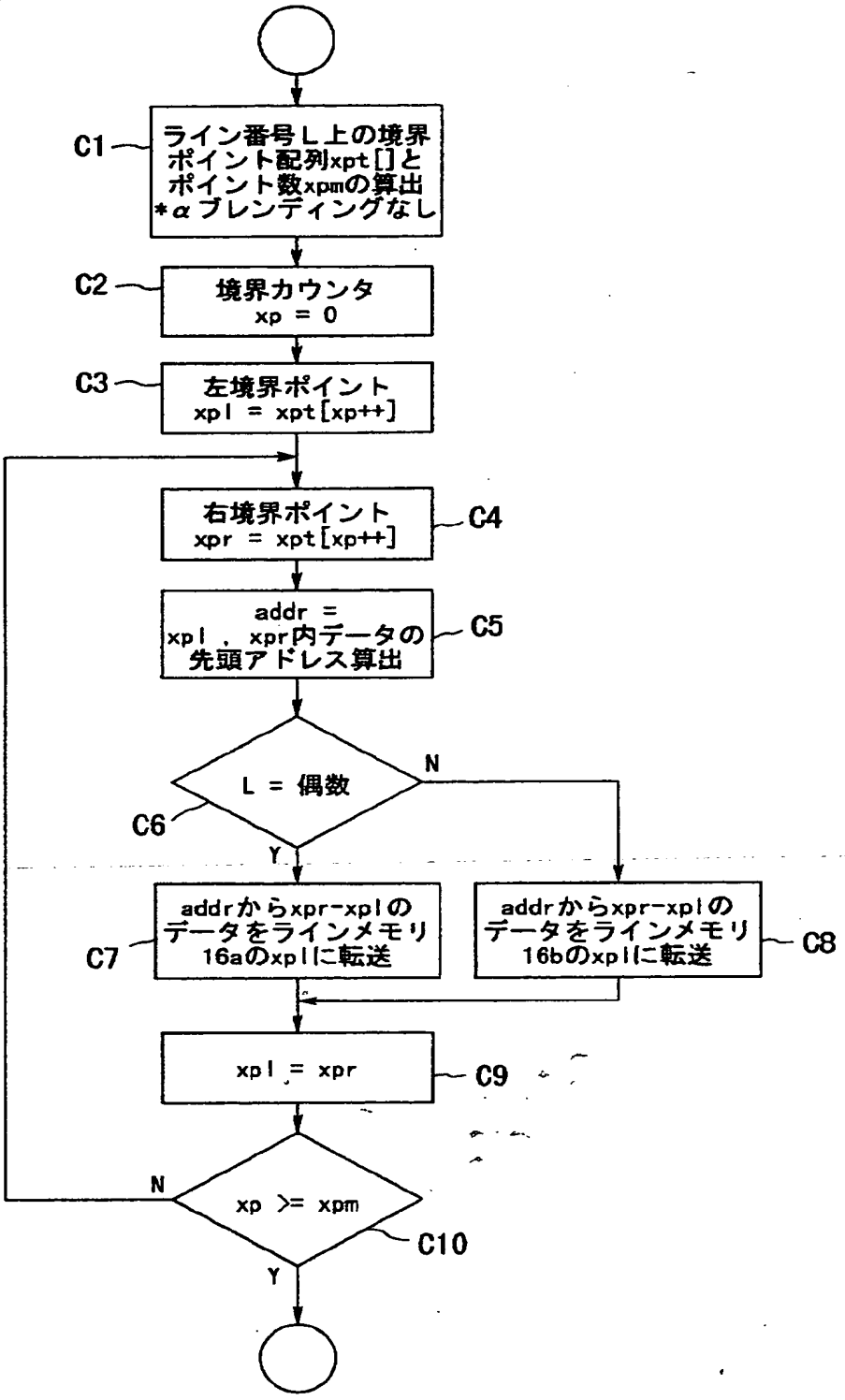
【図８】


他のデータ
back_addr
背景データ
curs_addr
カーソルデータ
win1_addr
ウインドウ１データ
win2e_addr
ウインドウ２偶数データ
win2o_addr
ウインドウ２奇数データ
他のデータ

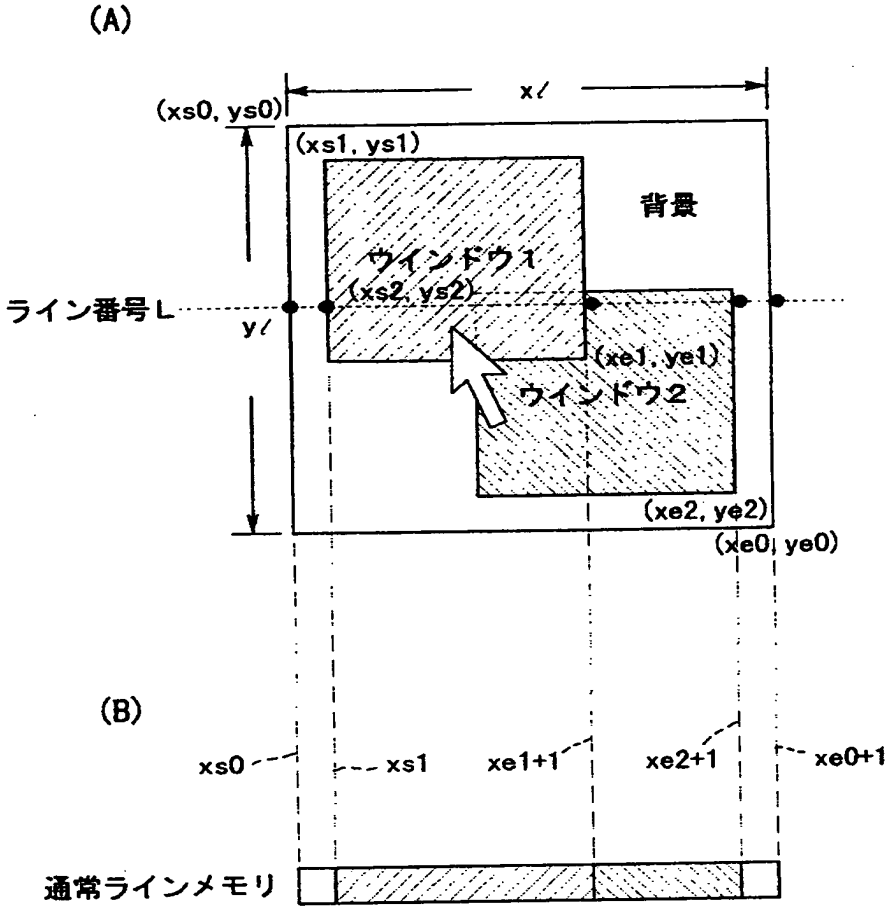
【図９】


START
メインメモリ上の
ウインドウデータを
内部メモリに転送
B1
yl=Y方向座標サイズ
B2
L =
次に表示する
ライン番号
B3
αﾌﾞﾚﾝﾃﾞｨﾝｸﾞは
ない
B4
N
Y
同期用ウエイト
B8
通常ライン転送
B5
αブレンディング
ライン転送
B6
(L+1) >= yl
B7
N
Y
END

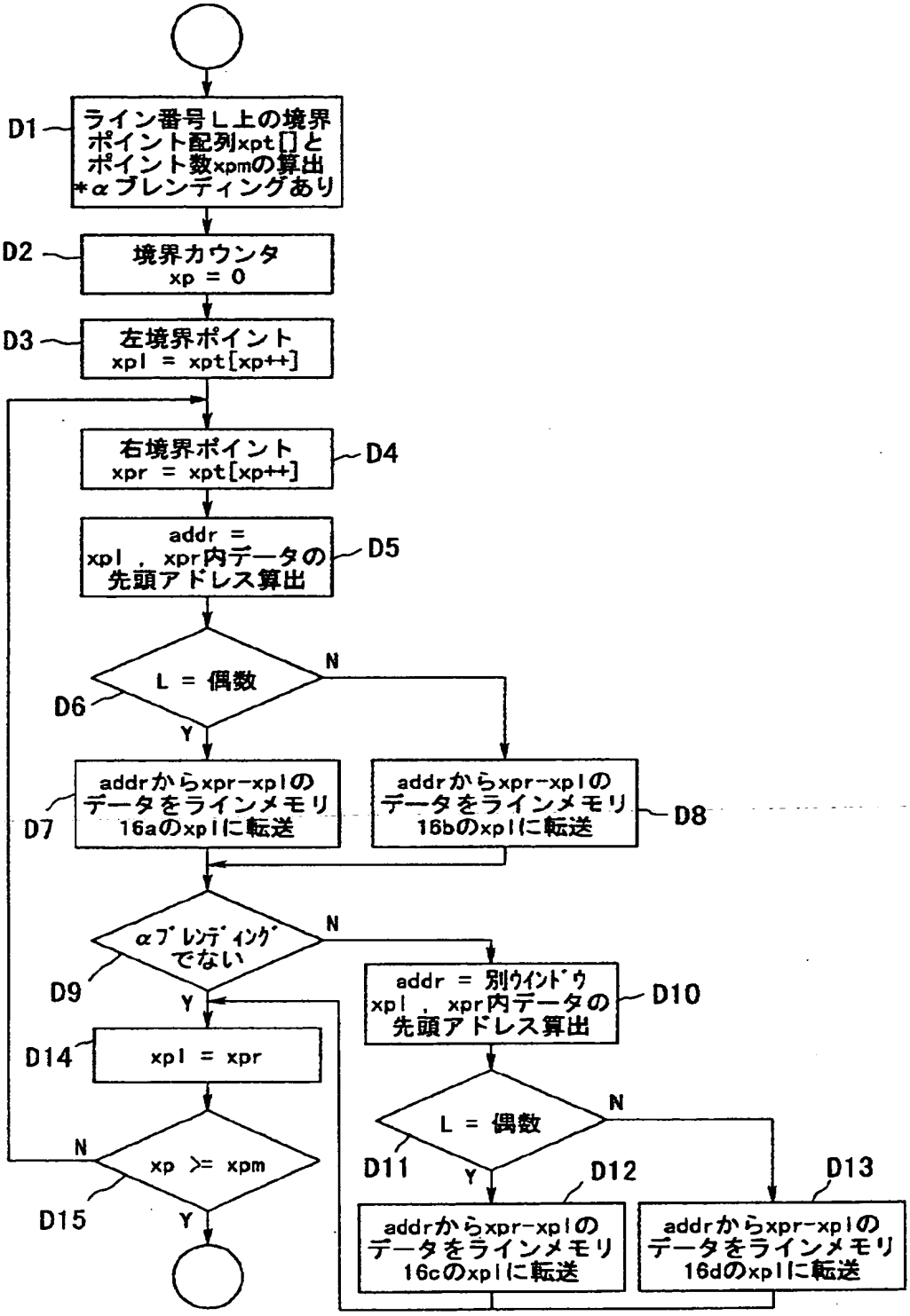
【図１０】


C1
ライン番号L上の境界ポイント配列xpt[]とポイント数xpmの算出 *αブレンディングなし
C2
境界カウンタ xp = 0
C3
左境界ポイント xpl = xpt[xp++]
C4
右境界ポイント xpr = xpt[xp++]
C5
addr = xpl , xpr内データの先頭アドレス算出
C6
L = 偶数
N
Y
C7
addrからxpr-xplのデータをラインメモリ16aのxplに転送
C8
addrからxpr-xplのデータをラインメモリ16bのxplに転送
C9
xpl = xpr
C10
xp >= xpm
N
Y

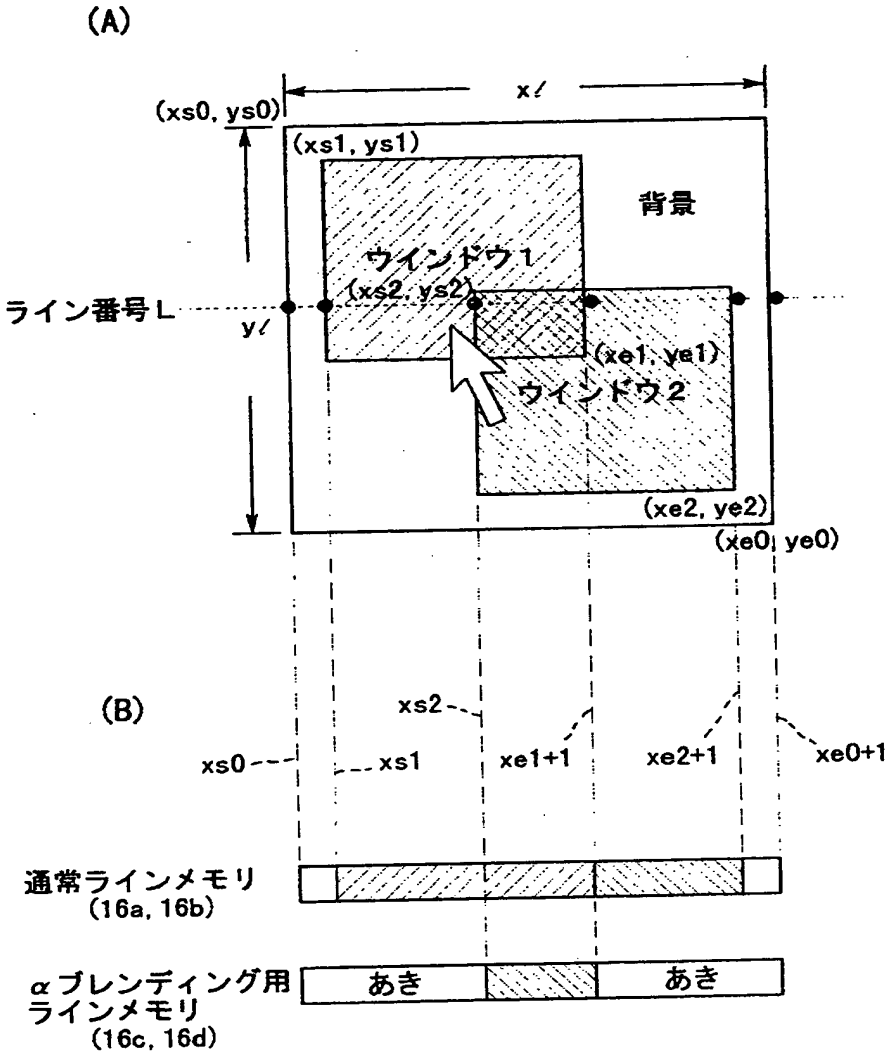
【図１１】


(A)
x
(xs0, ys0)
(xs1, ys1)
背景
ウインドウ1
(xs2, ys2)
ライン番号L
y
(xe1, ye1)
ウインドウ2
(xe2, ye2)
(xe0, ye0)
(B)
xs0
xs1
xe1+1
xe2+1
xe0+1
通常ラインメモリ

【図１２】


D1
ライン番号L上の境界ポイント配列xpt[]とポイント数xpmの算出 *αブレンディングあり
D2
境界カウンタ xp = 0
D3
左境界ポイント xpl = xpt[xp++]
D4
右境界ポイント xpr = xpt[xp++]
D5
addr = xpl , xpr内データの先頭アドレス算出
D6
L = 偶数
N
Y
D7
addrからxpr-xplのデータをラインメモリ16aのxplに転送
D8
addrからxpr-xplのデータをラインメモリ16bのxplに転送
D9
αブレンディングでない
N
Y
D10
addr = 別ウインドウ xpl , xpr内データの先頭アドレス算出
D11
L = 偶数
N
Y
D12
addrからxpr-xplのデータをラインメモリ16cのxplに転送
D13
addrからxpr-xplのデータをラインメモリ16dのxplに転送
D14
xpl = xpr
D15
xp >= xpm
N
Y

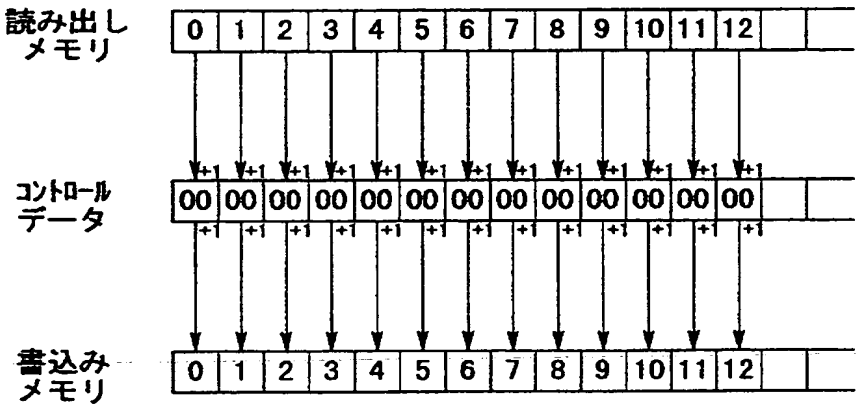
【図１３】


(A)
x
(xs0, ys0)
(xs1, ys1)
背景
ウインドウ1
(xs2, ys2)
ライン番号L
y
(xe1, ye1)
ウインドウ2
(xe2, ye2)
(xe0, ye0)
(B)
xs2
xs0
xs1
xe1+1
xe2+1
xe0+1
通常ラインメモリ
(16a, 16b)
αブレンディング用
ラインメモリ
(16c, 16d)
あき
あき

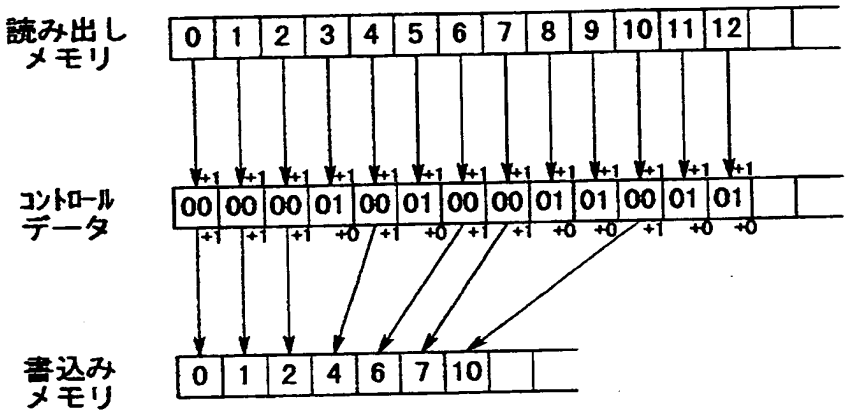
【図１４】

コントロールデータ	動作状態	処理後の各カウンタの動作		
		読出しカウンタ	書込みカウンタ	書込み動作
００	ノーマル	＋１	＋１	する
０１	縮小	＋１	＋０	しない
１０	拡大	＋０	＋１	する
１１	スキップ	＋０	＋１	しない

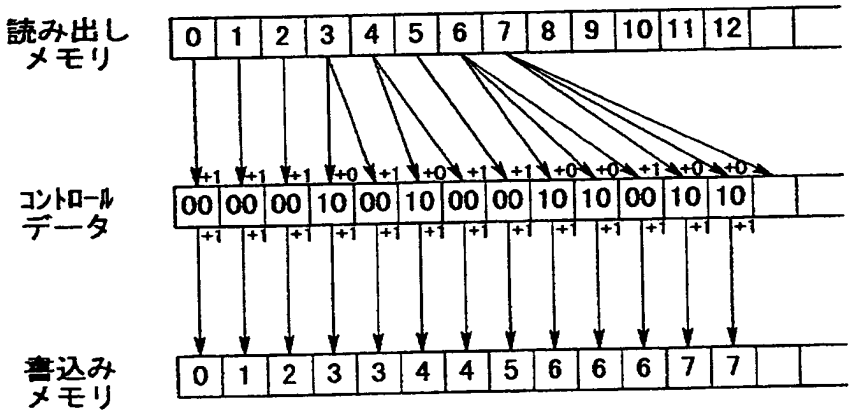
【図１５】



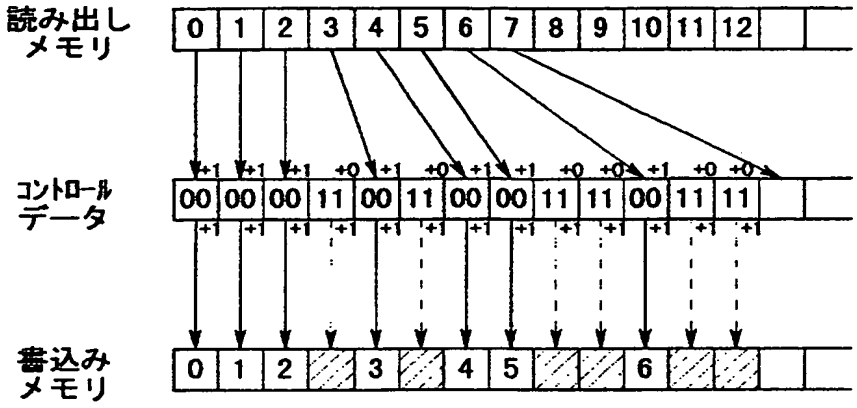
【図１６】


読み出し
メモリ
0 1 2 3 4 5 6 7 8 9 10 11 12
コントロール
データ
00 00 00 01 00 01 00 00 01 01 00 01 01
書込み
メモリ
0 1 2 4 6 7 10

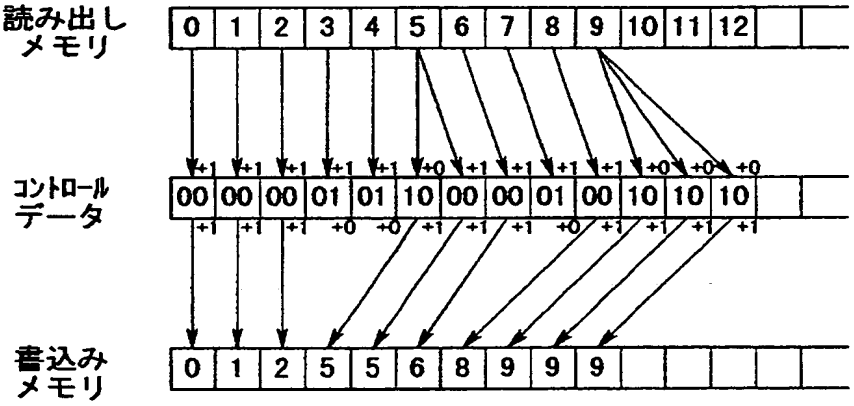
【図１７】


読み出し
メモリ
0 1 2 3 4 5 6 7 8 9 10 11 12
コントロール
データ
00 00 00 10 00 10 00 00 10 10 00 10 10
書込み
メモリ
0 1 2 3 3 4 4 5 6 6 6 7 7

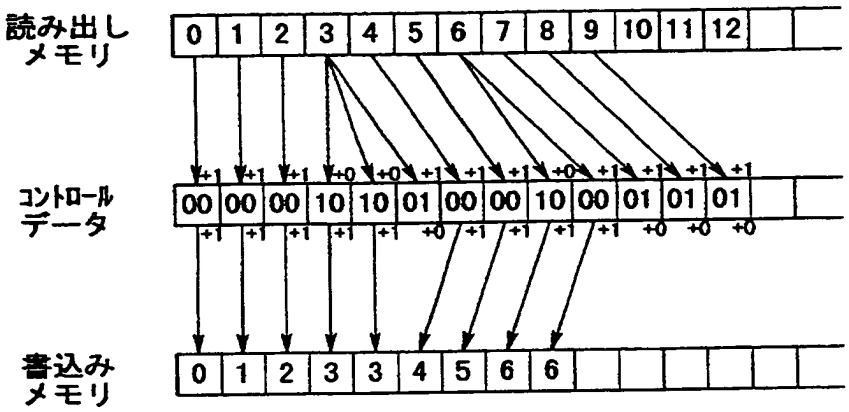
【図１８】


読み出し
メモリ
0 1 2 3 4 5 6 7 8 9 10 11 12
コントロール
データ
00 00 00 11 00 11 00 00 11 11 00 11 11
書込み
メモリ
0 1 2 3 4 5 6

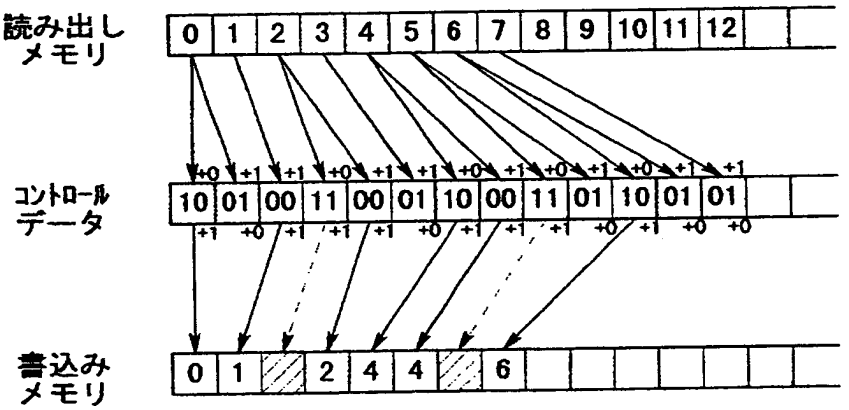
【図１９】


読み出し
メモリ
0 1 2 3 4 5 6 7 8 9 10 11 12
コントロール
データ
00 00 00 01 01 10 00 00 01 00 10 10 10
書込み
メモリ
0 1 2 5 5 6 8 9 9 9

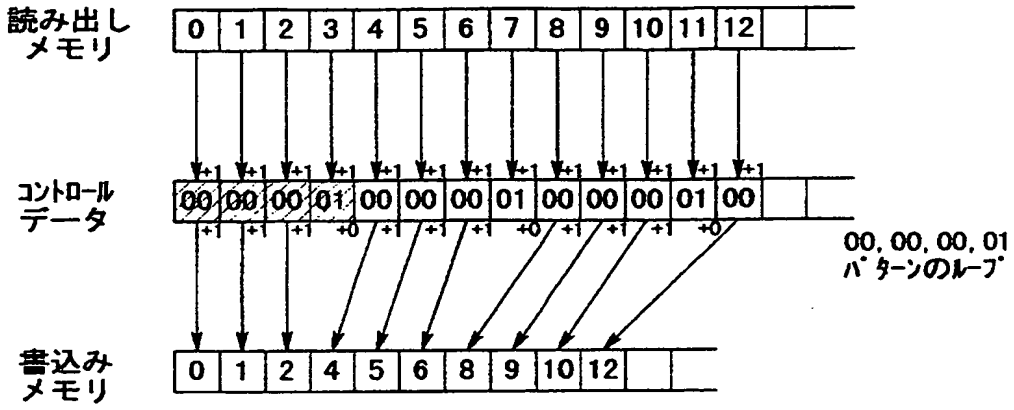
【図２０】


読み出し
メモリ
0 1 2 3 4 5 6 7 8 9 10 11 12
コントロール
データ
00 00 00 10 10 01 00 00 10 00 01 01 01
書込み
メモリ
0 1 2 3 3 4 5 6 6

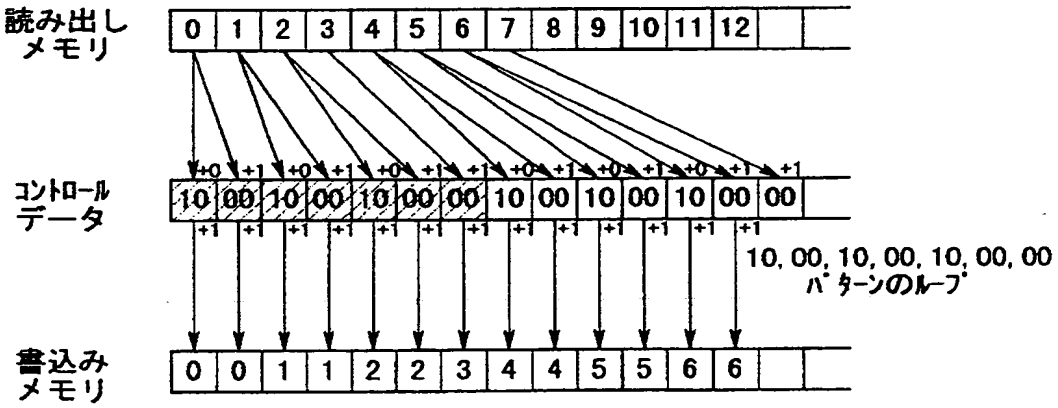
【図２１】


読み出し
メモリ
0 1 2 3 4 5 6 7 8 9 10 11 12
コントロール
データ
10 01 00 11 00 01 10 00 11 01 10 01 01
書込み
メモリ
0 1 2 4 4 6

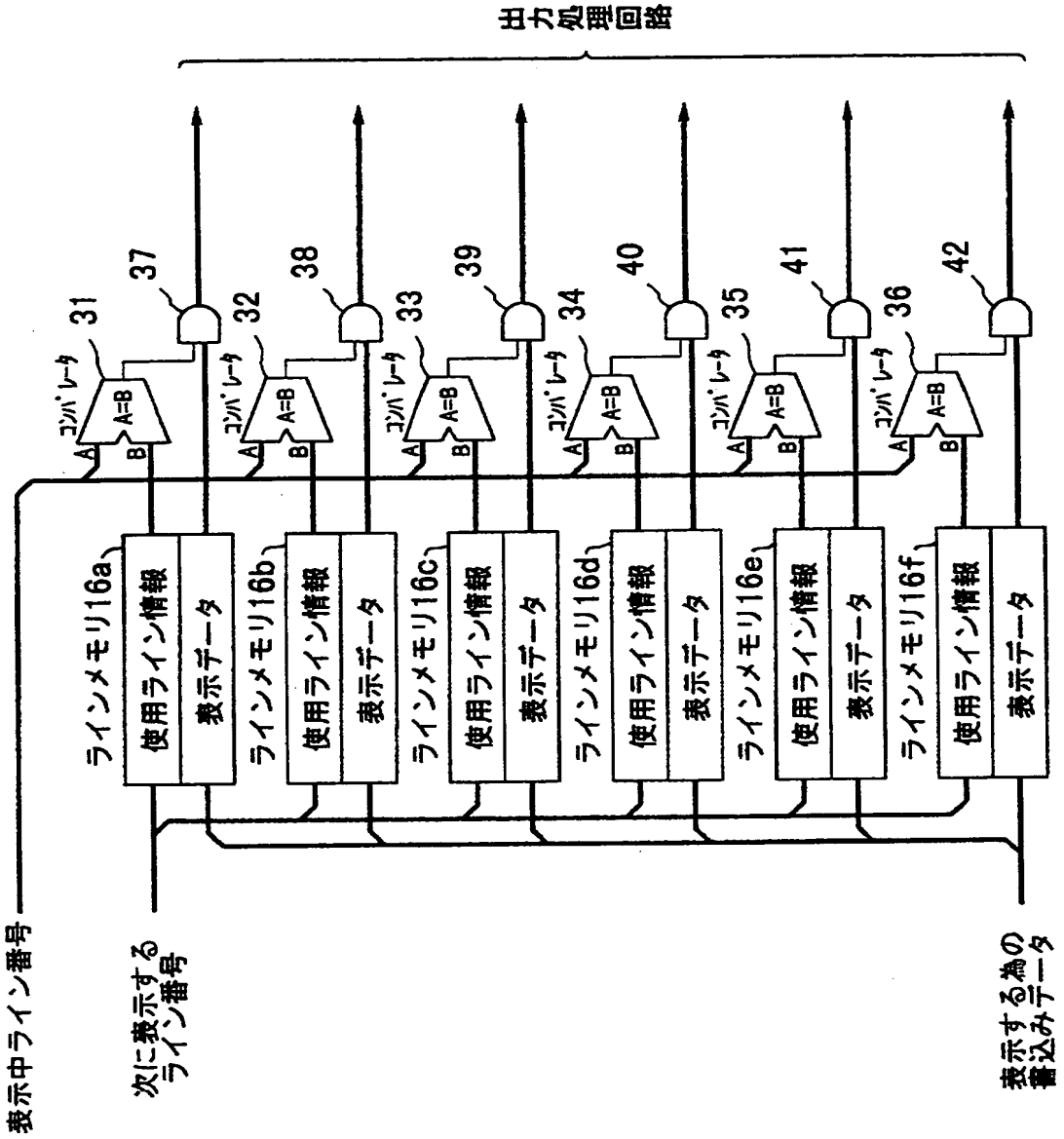
【図２２】


読み出し
メモリ
0 1 2 3 4 5 6 7 8 9 10 11 12
コントロール
データ
00 00 00 01 00 00 00 01 00 00 00 01 00
00, 00, 00, 01
パターンのループ
書込み
メモリ
0 1 2 4 5 6 8 9 10 12

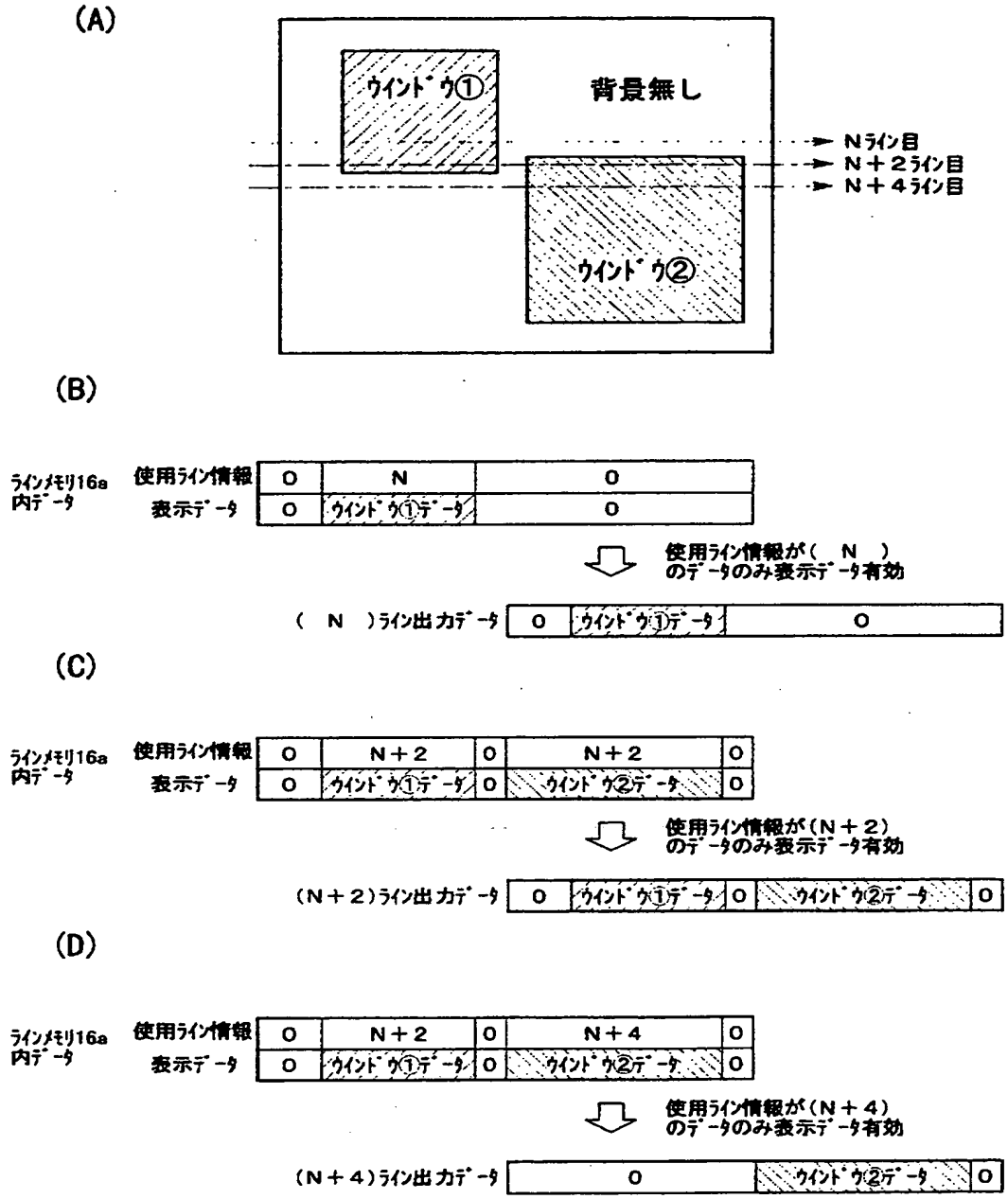
【図２３】


読み出し
メモリ
0 1 2 3 4 5 6 7 8 9 10 11 12
コントロール
データ
10 00 10 00 10 00 00 10 00 10 00 10 00 00
10, 00, 10, 00, 10, 00, 00
パターンのループ
書込み
メモリ
0 0 1 1 2 2 3 4 4 5 5 6 6

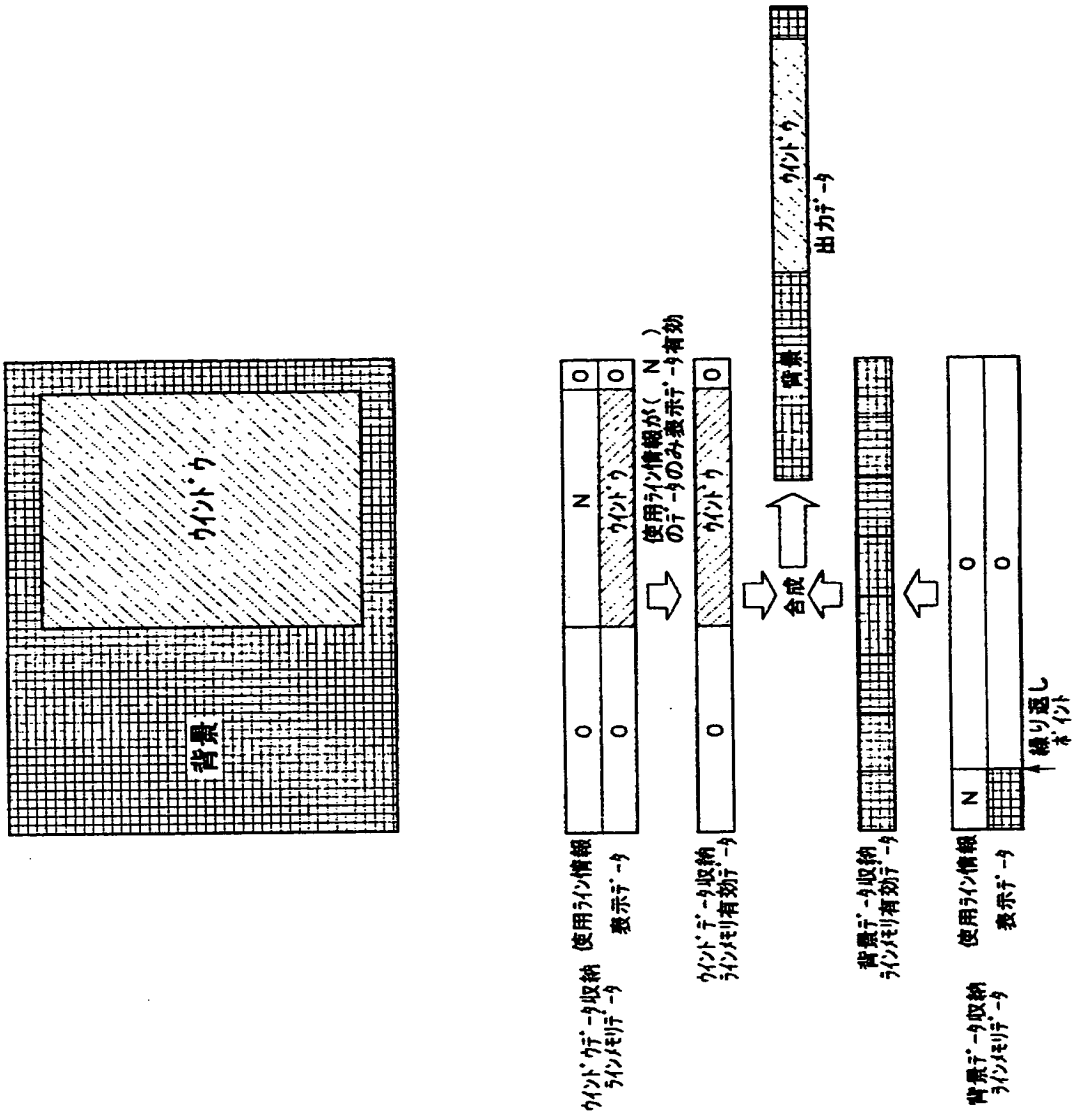
【図２４】


出力処理回路
31
32
33
34
35
36
37
38
39
40
41
42
コンパレータ
A
B
A=B
ラインメモリ16a
ラインメモリ16b
ラインメモリ16c
ラインメモリ16d
ラインメモリ16e
ラインメモリ16f
使用ライン情報
表示データ
表示中ライン番号
次に表示するライン番号
表示する為の書込みデータ

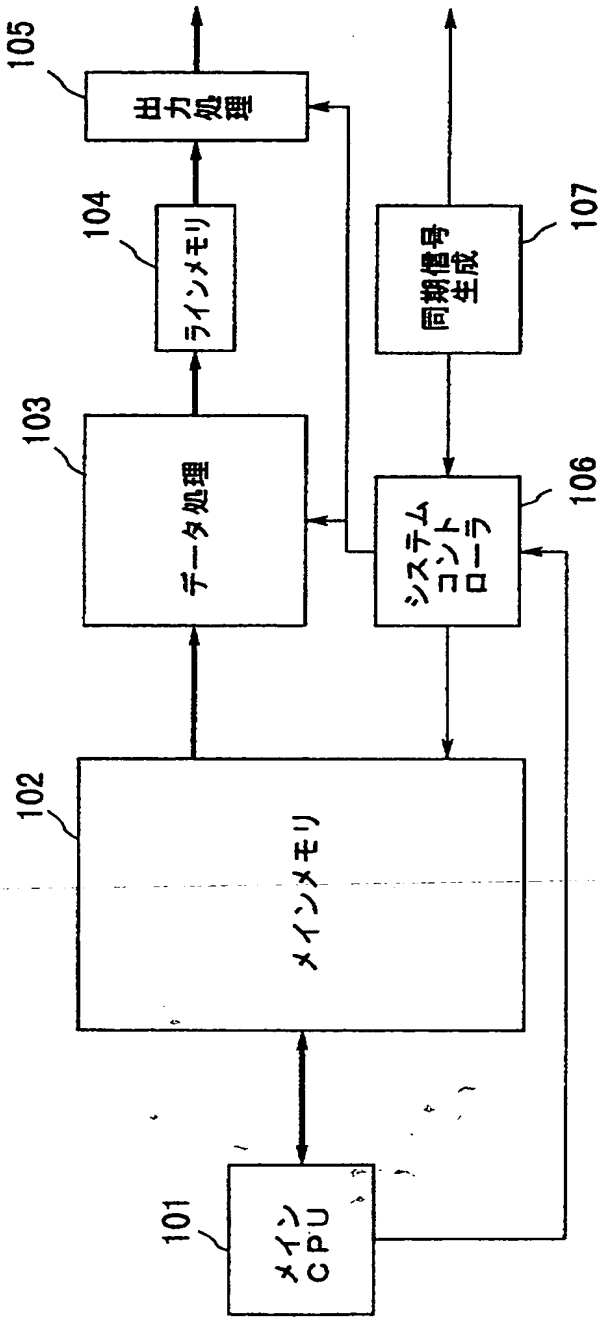
【図２５】


(A)
ウインドウ①
背景無し
Nライン目
N＋2ライン目
N＋4ライン目
ウインドウ②
(B)
ラインメモリ16a 内データ
使用ライン情報
表示データ
0
N
0
0
ウインドウ①データ
0
使用ライン情報がｎ
のデータのみ表示データ有効
（ N ）ライン出力データ
0
ウインドウ①データ
0
(C)
ラインメモリ16a 内データ
使用ライン情報
表示データ
0
N＋2
0
N＋2
0
0
ウインドウ①データ
0
ウインドウ②データ
0
使用ライン情報が（N＋2）
のデータのみ表示データ有効
（N＋2）ライン出力データ
0
ウインドウ①データ
0
ウインドウ②データ
0
(D)
ラインメモリ16a 内データ
使用ライン情報
表示データ
0
N＋2
0
N＋4
0
0
ウインドウ①データ
0
ウインドウ②データ
0
使用ライン情報が（N＋4）
のデータのみ表示データ有効
（N＋4）ライン出力データ
0
ウインドウ②データ
0

【図２６】


ウインドウ
背景
0
N
0
0
ウインドウ
使用ライン情報が（N）のデータのみ表示データ有効
ウインドウデータ収納ラインメモリデータ
使用ライン情報
表示データ
0
0
ウインドウ
ウインドデータ収納ラインメモリ有効データ
合成
背景
ウインドウ
出力データ
背景データ収納ラインメモリ有効データ
0
0
N
繰り返しポイント
背景データ収納ラインメモリデータ
使用ライン情報
表示データ

【図２７】


105
出力処理
104
ラインメモリ
107
同期信号生成
103
データ処理
106
システムコントローラ
102
メインメモリ
101
メインCPU

【書類名】　　要約書

【要約】

【課題】　表示データを収納するメモリ空間を必要なものだけとし、表示のためのメモリアクセス回数を押さえて処理を高速化できるとともに、主制御部の負担を軽減することができるプログラマブル表示装置を提供することである。

【解決手段】　メインＣＰＵ１１、プログラムや表示データやその他のデータを記憶するメインメモリ１２、メインメモリ１２の表示データをディスプレイ表示のデータ形式に変換する処理を行うデータ処理回路１３、変換処理された表示データを記憶する表示メモリ部１４、表示データを画面に出力するための処理を行う出力処理回路１７、メインメモリ１２へのデータアクセスを行うＤＭＡ１８、プログラムメモリ１９、データメモリ２０、プログラムメモリ１９やデータメモリ２０に記述された命令・データを解釈し、それに従っておもに表示データの転送等を行う表示プロセッサー２１、同期信号生成回路２２、とから構成される。

【選択図】　　図１

【書類名】　職権訂正データ
【訂正書類】　特許願

＜認定情報・付加情報＞
【特許出願人】
【識別番号】　000005049
【住所又は居所】　大阪府大阪市阿倍野区長池町２２番２２号
【氏名又は名称】　シャープ株式会社
【代理人】　申請人
【識別番号】　100069534
【住所又は居所】　東京都千代田区永田町２丁目１４番２号　山王グランドビルヂング３階３１７区　藤本特許法律事務所
【氏名又は名称】　藤本　博光

出　願　人　履　歴　情　報

識別番号　　〔０００００５０４９〕

１．変更年月日　　１９９０年　８月２９日
［変更理由］　新規登録
住　所　　大阪府大阪市阿倍野区長池町２２番２２号
氏　名　　シャープ株式会社